Aalto University
School of Electrical
Engineering

Ignacio García Gosálbez

Demand Control tool for houses with Thermal Energy Storage Systems

Final Project of the degree of Industrial Engineer in the programme of Electrical Engineering.

Supervisor: Matti Lehtonen

August 20, 2014

Author:	Ignacio García Gosálbez
Department:	Department of Electrical Engineering
Major Subject:	Demand Response
Research Project:	Smart Grids and Energy Markets
Title:	Demand Control Tool for houses with Thermal Energy Storage Systems.
Supervisor:	Prof. Matti Lehtonen
Professorship:	Power Systems & High Voltage Engineering

Abstract

The goal of this thesis is to develop a load control of houses with Thermal Energy Storage systems in order to avoid the appearance of uncontrolled consumptions during peak hours. In the Nordic countries, the installation of energy storage in households is a common practice: these systems are used to decouple the heating and the electric consumption of the household, enabling the consumer to shift the most electric demand from on-peak to off-peak hours.

These kind of electric customers are subscribed to specific electricity tariffs. These tariffs are composed by two levels of electricity price, being cheap to consume at night time and expensive during daytime: they are known as “Time of Use” tariffs. Houses with full storage capacity satisfy their entire heating demand by consuming during “Time of Use” hours; however, houses with partial storage capacity, which own a smaller storage tank, exhibit extra consumptions during daytime out of “Time of Use” hours when their storing capacity is not big enough to supply the heating demand of the house. These extra consumptions are a priori unexpected, and their appearance may cause economic losses to the electricity distributor.

Finland buys electricity in the Nord Pool Spot, which runs the largest electrical market in the world. The daily evolution of the electricity price is known a day-ahead in the called Elspot market: this allows distributors to create efficient and economic daily consumption plans for different customers. In this thesis it is presented a tool to create a consumption plan for houses with Thermal Energy Storage by analysing the daily Elspot price evolution.

The starting data are electric records obtained by metering at distribution level. These records belong to houses located in four regions of Finland. Among these houses, some of them exhibit similar consumption trends to a Thermal Energy Storage system: they have been clustered and studied independently. Once the most relevant parameters of each house are estimated, such as the boiler power and storage capacity, it is applied a mathematical model in order to simulate the hourly evolution of the storage level. A linear model of heating consumption as a function of the outdoor temperature is presented in this work; in addition, the historical deviations of consumption from the linear model are studied, and play an important role in the demand forecast proposed here. The model of hourly evolution of the storage level is used to develop a day-ahead control. Through the demand forecast, it is computed a feasible consumption level for the next day.

A control vector is created every day and given to the customer. This vector offers the most economic loading pattern that satisfies the heating demand, and allows the controller to estimate the actual storage level of the house; the loading pattern is generated by using the Stochastic Genetic Algorithm, an application of the Genetic Algorithm, widely used in optimization processes.

Finally, it is also presented a storage level updating tool that allows tracking the behaviour of the customer regarding to the control given.

Number of pages: V+61	**Keywords:** Time of Use, Thermal Energy Storage, Elspot price, storage tank, Stochastic Genetic Algorithm.

Autor:	Ignacio García Gosálbez
Departamento:	Departmento de Ingeniería Eléctrica
Tema Principal:	Respuesta de la demanda
Proyecto de Investigación	Redes Inteligentes y Mercados Energéticos
Título:	Herramienta de Control de la Demanda de viviendas con Sistemas de Almacenamiento de Energía Térmica.
Supervisor:	Prof. Matti Lehtonen
Cátedra	Sistemas de Potencia e Ingeniería de Alta Tensión

Resumen

El objetivo de este estudio es el desarrollo de un control de carga en viviendas con almacenamiento de energía térmica, con el fin de evitar la aparición de consumos inesperados durante las horas pico. En los países Nórdicos, la instalación de almacenamientos de energía es una práctica común: estos sistemas se usan para desacoplar las demanda de calefacción y la eléctrica en el hogar, permitiendo al consumidor mover gran parte de su demanda energética de horas pico a horas valle.

Este tipo de consumidor eléctrico está subscrito a unas tarifas eléctricas específicas. Estas tarifas están compuestas por dos niveles de precio de electricidad, siendo barato el consumo nocturno y más caro el consumo diurno: se las conoce como tarifas de "Tiempo de Uso". Las viviendas con almacenamiento completo satisfacen toda su demanda de calefacción consumiendo básicamente durante horas de "Tiempo de Uso"; sin embargo, los hogares con almacenamiento parcial, que poseen un tanque de capacidad reducida, exhiben consumos diurnos extra fuera de estas horas "Tiempo de Uso" cuando su capacidad de almacenamiento no es lo suficientemente grande para abastecer toda la demanda de calefacción de la casa. Estos consumos extra son a priori impredecibles, y su aparición puede suponer pérdidas económicas para el distribuidor eléctrico.

Finlandia compra la electricidad en el Nord Pool Spot, que controla el mayor mercado eléctrico del mundo. La evolución diaria del precio de la energía se conoce con un día de antelación en el llamado mercado Elspot: esto permite a los distribuidores crear planes de consumo diario económicos para diferentes clientes. En este trabajo se presenta una herramienta que crea un plan de consumo para casas con almacenamiento de energía térmica mediante el análisis de la evolución diaria del precio Elspot.

Los datos de partida son los registros eléctricos obtenidos mediante medición a nivel de distribución. Estos registros pertenecen a casas situadas en cuatro regiones de Finlandia. Entre estas casa, algunas exhiben tendencias de consumo similares a un sistema de almacenamiento de energía térmica: estas casas se han agrupado y estudiado independientemente. Una vez se han establecido los parámetros más relevantes de cada casa, como la potencia de la caldera o la capacidad de almacenamiento, se aplica un modelo matemático con el fin de simular la evolución horaria del nivel de almacenamiento. Un modelo lineal de consumo de calefacción como función de la temperatura exterior se presenta en este trabajo; además, se estudian las desviaciones históricas de consumo a partir de este modelo lineal, que desempeñan un papel importante en la predicción de demanda aquí presentada. El modelo de evolución horaria del nivel de almacenamiento se usa para desarrollar un control con un día de antelación. Mediante la predicción de la demanda, se calcula un consumo factible para el día siguiente.

Un vector de control se crea cada día, y es dado al cliente. Este vector ofrece el patrón de consumo más económico que satisface la demanda de calefacción, y permite al controlador estimar el nivel real de almacenamiento de la casa; el patrón de carga se genera mediante el uso del Algoritmo Genético Estocástico, una aplicación del Algoritmo Genético, ampliamente usado en procesos de optimización.

Finalmente, se presenta también una herramienta de actualización del nivel de almacenamiento que permite realizar un seguimiento del comportamiento del cliente con respecto al control dado.

Número de páginas: V+61	**Palabras Clave:** Tiempo de Uso, Almacenamiento de Enería Térmica, precio Elspot, tanque de almacenamiento, Algortmo Genético Estocástico.

Preface

The work for this Final Project was done at the Smart Grids and Energy Markets (SGEM) research group, on the Department of Electrical Engineering and Automation at the School of Electrical Engineering of the Aalto University, in Helsinki, Finland. The focus areas of research in power systems are distribution networks, their system solutions and reliability engineering, power system components´ load capacity and ageing, control and monitoring systems including computer applications, automation and communication solutions, end use analysis and power quality. The head of research group is Professor Matti Lehtonen, who has a solid experience of more than 20 years within the above themes. The research work is done on close cooperation with manufacturing and energy industry.

I am very grateful to my supervisor Prof. Matti Lehtonen for giving me the chance of joining this research group; his support and advices have been fundamental to me to keep on working every day with illusion. In addition, he has always tried to find the best way to help me in this adventure I want to start here, in Finland.

A very special mention to Diego, who has helped me from the very first moment I landed in Finland, always with a big smile and tones of patience and useful advices. Another mention to Alberto, who is always ready to improvise a pasta and gives me my daily dose of genuine southern spirit. To my lads Nicola and Simo, with whom I have shared countless saunas speaking about life.

Carlo, Tammo, and Magdalena, with whom I have been living for 9 months, and are the best company anybody can have. Uri and Pancho (the inseparable fellows) and our coffee sessions in Radioscience.

My friends in Spain David, Roberto and Piero, always ready to travel and to visit me wherever I am, and whose support and good humour never fail.

I would also like to express my gratitude to Jaakko Köhli and Ana Domingez, who are always available for any request about international students, and have always eased the bureaucratic part of my experience. And to the UPM for allowing students to all travel around the world to improve as professionals and as persons. I wish the next generations will enjoy this opportunity same as I have done.

Finally, my deepest appreciation goes to my family, for their endless help and because they represent the best example of self-improvement, re-inventing themselves to keep on rowing in life. To my father for his perseverance to never give up even if life knocks you down; to my mother for her understanding and concerns; to my sister for her support, and to my grandmother, who is simply the best.

And of course, the last words are for Evgeniia, who gives me the strength to wake up every morning full of energy to keep on fighting against the difficulties.

Otaniemi, August the 20th, 2014.

Ignacio García Gosálbez

Contents

Figures

Tables

Acronyms and Symbols

TES	Thermal Energy Storage
ToU	Time of Use
COP	Coefficient of Performance
GA	Genetic Algorithm
SGA	Stochastic Genetic Algorithm
FGRV	First Generation of Response Vectors
SGRV	Second Generation of Response Vectors
P_B	Boiler Power
C_S	Storage Capacity
$n_{l.h.}$	Number of loading hours
S_i	Storage level at hour "i"
S_0	Initial storage level at the beginning of the day
x_i	Component "i" of the response vector
w_i	Heat consumption at hour "i"
W_{day}	Total daily heat consumption
T	Temperature outdoors
$t_{A,B,C}$	12 am of days A, B and C consecutively.
t_i	Starting hour of Time of Use
t_f	Final hour of Time of Use
$x_{0,(A,B)}$	Percentage of initial storage level at t_i
$K_{Elspot,i}$	Elspot price at hour "i"
α	Price limiting parameter
dev	Deviation level from average consumption
lb	Lower bound
ub	Upper bound
$S_{change,(1,2)}$	Storage level automatic update inside the control

Chapter 1. INTRODUCTION

The improvement of the electric network's efficiency is nowadays one of the major targets to electric power companies. In particular, the unbalance between the generation and consumption causes instabilities to the network, leading to efficiency losses in the electric system. The control of Demand Response (DR) is seen as a feasible solution to stabilize this unbalance between both sides, reason why the so called Smart Grids are being developed in the last decades; in this way the generators can plan beforehand the energy production for the following day according to a forecasted demand.

A *Smart Grid* is a modernized electrical grid where an important information flow exists between energy suppliers and customers in order to understand, to predict and match their behaviours. Traditional methods of power supplying, by burning fossil fuels such as coal, gas, oil or peat, allow the generator to control accurately the amount of electricity to feed into the grid; nevertheless, the inclusion of new alternative sources such as renewable energies has led to introduce a stochastic behaviour also on the generation side: reason why all the elements that form the electric system have to be independently studied and modelled.

"*The network between the measurement devices and business systems allows collection and distribution of information to customers, suppliers, utility companies, and service providers*" [1]. The starting data to create consumption models in order to perform demand forecast are the electrical records, where the customer's consumption habits appear reflected; the tool used to record the electric consumption is the *meter*.

Metering

"*Like transmission systems, distribution systems rely on cables, poles, and substations - they just operate at lower voltages. A major distinction is that the electricity is consumed at the end of the distribution circuit. Uses can range from heating food to lighting offices to welding heavy equipment; different classes of end-users have different needs and differing patterns of use. The last step in a utility's distribution system is the meter.*" [2]

Metering has traditionally been done in networks at transmission level for security reasons: "*while electric meters have been around for over 100 years (Edison invented one for his lighting system), they have seldom provided much information. Usually just a simple measure of the amount of electricity used"* [2]. At transmission level, the demand can be forecasted in an easier way as it belongs to a set of customers: a population, in set, usually follows the same consumption routines. However a single consumer demand is more complicated to predict since it can vary dramatically from one day to another. This is an issue for the distribution companies, which are in charge of supplying this service.

"*Recent developments make it possible for electricity suppliers to collect much more detailed information from a new generation of meters.*" [2] Thus consumption metering is being lately implemented in distribution networks, allowing a more efficient tracking of lower levels of consumption and providing a better service maintenance close to the customer. Particularly in Finland, "*developing the energy infrastructure towards smart grids is a longer term national objective, with smart meter rollout being one of the first steps*". [3]

Heating consumption groups

Based on official studies about the 80% of the total domestic energy consumption is used in heating of residential buildings, whereas household appliances consumption only represent a 16% [4]. "*In Finland, the annual electricity consumption of households varies considerably, mainly depending on whether the household has electric heating*" [5]. "*Most common domestic heating systems in Finland are electric heaters, heat pumps, radiators or under floor and ceiling heaters*" [6]. In terms of electric consumption, there are two main types of consumers in Finland from the distribution company's point of view:

1) Customers that do not use electricity for heating, where household applications represent the main consumption. This group is composed by district heating subscribers (whose heating demand is supplied by a district heating company), and customers with an alternative heating system such as wood, oil or gas boilers. Their heating consumption is not reflected on the meter reading. In 2007, "*District heating was the most common heating method in apartment houses (76%) and row houses (43%)".* [6]

2) Customers with electric heating system: this group exhibits a higher electric demand compared to the first group since electric heating consumption is included on the record, which represents around the 80% of energy consumption in a household. In this case, the record shows a combination of both heating and appliances' consumption. "*In the end of 2007, 42% of detached houses and 32% of row houses had electric heating"* [6]. The common electric space heating systems in Finland are as follows:

    - Direct electric heating
    - Storage electric heating
    - Heat pump

Storage electric heating is the group of interest in this work. These systems work by storing the heat energy inside materials with a big thermal inertia; they are commonly named *Thermal Energy Storage* (TES) on literature reviews[1]. In these houses "*the energy is supplied to the TES (charging mode) where it is stored, then drawn from the TES (discharging mode) and used at a later time*". [7] There are different widely spread technologies depending on their storing medium:

1) Electric Thermal Storage Heaters, that consist on well insulated clay bricks or other ceramic materials heated to a high temperature with electricity, and well insulated to release heat over a number of hours.
2) Heat Storage in water tanks, widely used in Scandinavia [8].

[1] They are also named Seasonal Thermal Energy Storage (STES) when they have cooling capacity.

Houses with Thermal Energy Storage system

These houses are equipped with an electric boiler and a Thermal Energy Storage (TES) tank connected to the heating circuit. The most common TES system in the Nordic countries is a water storage tank: "*In water circulation heating systems heat is stored by means of a heating device into the water of the thermal storage tank. The thermal storage tank is equipped with an efficient heat storage and discharge technique, and the energy is consumed from the tank to the heating of the building or domestic water"* [9]. The space heating circuit is typically composed by a single closed circuit of pipes that connects the tank to the heaters along the house. *"To replace that hot water, cold water* (from the same closed circuit) *enters the bottom of the tank, ensuring that the tank is always full"* [10]. These systems are also used for tap water supply by extracting the heating capacity of the tank's water through heat exchangers to a secondary circuit, from which hot tap water is drawn-off. "*It operates by releasing hot water from the top of the tank when you turn on the hot water tap"* [10].

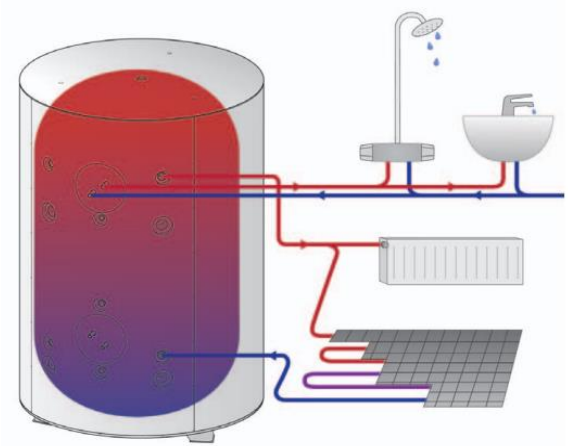


Figure 1 - TES system [9]

"*TES are widely recognized as means for decoupling electricity production and demand.* […] *TES systems have also proved useful for shifting electrical loads from peak to off-peak hours, becoming a powerful demand-side management tool"* [7]. These systems load electricity from grid in off-peak to charge the tank, and supply the heating demand of the house along the whole day with the stored energy.

"TES systems are operated in two modes: full storage and partial storage. Full storage systems, also known as load shifting systems are designed to meet all on-peak heating loads from storage. Partial storage systems meet part of the heating load from storage and part directly from the boiler during the on-peak period" [11]. When planning a full storage system, the tank's capacity is typically overdimensioned in order to ensure the heating supply even if very low temperatures are reached; thereby the boiler only loads during *ToU* hours. "*This type of system results in larger and, therefore, more expensive boiler and storage units compared to partial storage systems. However, full storage also captures the greatest savings possible by shifting the most electric demand from on-peak to off-peak".* [11]

"*In general, partial storage systems meet part of the heating load from storage and part directly from the boiler during the on-peak period*" [11]. This is the problem that partial storage systems entail to distributors: the part of the heating load met during the on-peak period is unpredictable and can match the most expensive electricity price of the day. This supposes efficiency losses since it can overload the grid, and can also bring economic losses to the distribution company.

TES systems and the distribution company

The distribution company allocates electricity from power plants to customers, and is *"responsible for the functionality and modernisation of its electricity networks, measuring its customers' electricity consumption and submitting energy data to electricity retailers".* [12] *"Electricity retailers provide fixed prices for electricity to their customers and manage the risk involved in purchasing electricity from spot markets or electricity pools"* [13]. Elspot is the day-ahead market of electrical energy in the Nordic countries, integrated inside the Nord Pool Spot that runs the largest electrical market in the world [14]. The daily Elspot price profile is known one day ahead, allowing the distributor to select the most profitable hours to offer to its customers.

"Customers can have a more efficient system and save money if they take advantage of different electricity prices during peak and off-peak hours; and utilities can spread the demand over the whole day, enabling a more efficient use of existing generating capacity and reducing the need for new capacity" [7]. *"Nearly 79000 consumers in Finland are equipped with storage electrical heating"* [15]. These customers are often subscribed to *Time of Use* (*ToU*) tariffs on which the electricity price varies depending on the time-of-day when the service is provided: during off-peak hours (from 10 pm to 7 am of the following day[2]) the electricity is cheap, whereas during daytime (from 7 am to 10 pm) the electricity price is higher.[3] *Night time prices are often as 50% lower than day time pricing, hence giving incentive to customer to store heat during night time and allowing it to coast for the rest of the day".* [15]

Distribution companies ignore whether a house has a TES system or not; but it is found between *ToU* subscribers that some records exhibit peculiar tendencies in terms of consumption profile: some of them show high levels of consumption during night time when most of the household appliances are off; moreover along certain periods of the year, especially in winter, this consumption during *ToU* represents almost the 100% of the daily electric demand. If also is taken into account that statistically around the 80% of the energy consumption is destined for heating residential buildings in Finland [4], there are enough evidences to interpret that this night time consumption is due to a TES system in charging mode.

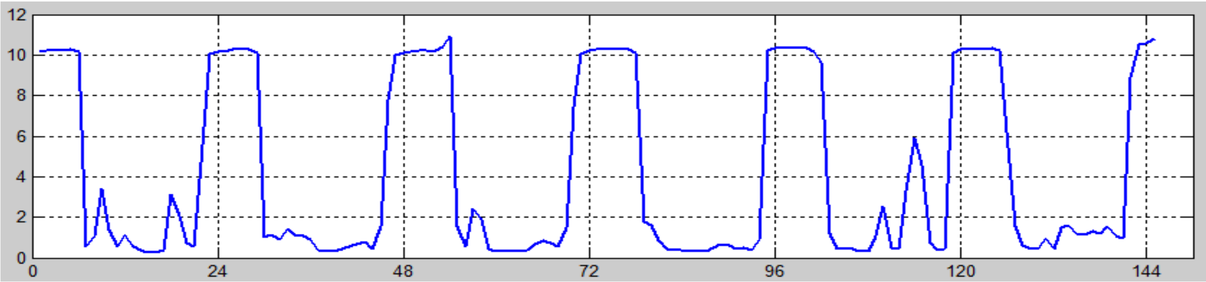


Figure 3- Sample of a house from Jyväskylä. The difference between daytime and night consumptions can be appreciated

The picture shows this tendency found in records: high night consumptions (when the TES is in charging mode) and lower demand during daytime (when the TES is discharging energy to the heating system).

[2] Reference taken from [40]

[3] From this point, the acronym *ToU* will be referred as the time period between 10 pm to 7 am.

Aim: development of a TES control-tool

"*The restructuring energy marketplace presents challenges and opportunities for energy providers and users alike. TES can play an increasingly important role in addressing some of the challenges and in creating value among the opportunities*". [16]

The distribution company can improve the system's efficiency and provide a better supply service by knowing the customers' consumption habits; moreover, the demand can be spread to avoid overloads at distribution level. The motivation of this study is to develop a load control of TES systems in order to avoid the appearance of uncontrolled consumptions during peak-price hours. These unexpected consumptions appear regardless the hourly Elspot price when extra heat is needed, especially in systems with partial storage. The development of control tools would allow distribution companies to avoid unexpected peak loads while minimizing economic losses.

The efficiency of TES systems to shift electricity demand is studied in several literature reviews [7] [17] [18] [19] [20] [21] by addressing the issue from the customer's point of view: in this way the component details of the heating system are known beforehand and can be used as starting data. Furthermore, the problem lies in the lack of this information in this present work: the only available data to the distribution company is the metered record of total consumption.

Thereby, a historical consumption analysis is the starting point of this study: the data are electric records of metered customers from different regions of Finland, such as Pirkanmaa, Uusimaa, Tavastia Proper and Central Finland.[4] The meter gives a reading per hour, so the daily record is composed by 24 values: this same structure is kept to perform a 24-component control vector to give to the customer one day in advance.

The electric record encompasses the electric heating demand as well as the household appliances consumptions, so the first step is to isolate the consumption allocated for heating purposes. In this way an estimation of the *Storage Capacity* and *Boiler Power* can be performed: these two values are key data to plan the house control. It has been found that during certain days of the year some houses experiment high consumptions during daytime, thus modifying their regular consumption trend: the program presented here finds these high extra consumptions during daytime, whose appearance might mean the heat storage is depleted.

The main goal is to create a control vector that works as a time switch to the boiler; thus, the electric demand can be shifted by controlling the boiler's loading hours. The 24-component control vector follows the architecture of a binary sequence, where a "0" tells to switch off the boiler and a "1" offers a loading hour.

In this work it is also presented an application of the standard Genetic Algorithm (GA), the *Stochastic Genetic Algorithm*, developed to improve the convergence speed of the standard GA and to shape the final control vector to provide to the customer. The GA is widely used in literature reviews as an optimization tool in DR control [22] [23] [24] [25] [26].

[4] These regions have been identified thanks to the customers' post codes.

Chapter 2. SYSTEM PARAMETERS AND MATHEMATICAL MODEL

The main technical specifications to define a TES system are the *Boiler Power* ($P_B, in\ kW_{elec}$) and the *Storage Capacity* ($C_S, in\ kWh_{heat}$): the knowledge of these two values allows the controller to simulate the TES behaviour. In this study, these specifications are estimated by performing an analysis of the historical consumption record.

Before analysing the metered consumption, it needs to be understood how a TES system operates: thereby, the boiler consumption can be isolated from the other household appliances' consumptions. A typical electric boiler for a full capacity system presents an output power from 1 kW-24 kW depending on the space that wants to be heated [27] [28] [29] [30], with a COP (*Coefficient of Performance*) close to unity[5].

Inside the available records in this study, it can be seen that the majority of these houses exhibit consumption peaks between 9 kWh-16 kWh. These peaks are very high compared to the average consumption of other household appliances (typically below 2.5 kW): this can be seen at consumption lists [31] [32] of the most common household appliances. In addition, this peak consumption remains constant along several *ToU* hours in the records, which suggests that this value is the actual P_B of the TES system.

As it has been found on the literature, the power of an electric boiler is typically an integer value (*e.g.* 14 kW); P_B is computed as the mode of the integer values of all the consumption peaks historically metered. The P_B represents the amount of heat input to the storage in every loading hour. In houses with *ToU* tariffs the boiler loads several hours during night time. In load levelling versions of control, the boiler always works at full capacity.

The number of charging hours depends on the *Storage Capacity* ($C_S, in\ kWh_{heat}$). The boiler switch is controlled by a thermostat located inside the storage that owns a temperature reference; the thermostat switches off the boiler when the temperature inside the storage goes over the reference value. The estimation of C_S is computed as the number of loading hours times P_B in a *critical day*[6].

$$C_S\ [kWh_{heat}] = n_{l.h}[h] \cdot P_B[kW_{elec}] \cdot COP_B \left[\frac{kW_{heat}}{kW_{elec}}\right] \tag{1}$$

The C_S and the *number of loading hours* ($n_{l.h.}$) vary from a system with partial storage to another with full storage capacity. In houses with full storage, where the C_S is high, the load is constant along several consecutive *ToU* hours, whereas houses with partial storage the load remains constant only few consecutive hours before the boiler switches off. This behaviour pattern is used to estimate both technical specifications for every customer.

[5] A typically value of a boiler COP can vary between 0.88-0.99 [41] [42], meaning that in one hour it is approximately obtained 1 kWh_{heat} per kWh_{elec} consumed from grid.
[6] The concept of critical day is introduced on the following page.

TES detection tool

The first step is to detect which houses are candidates to own a TES system. For this purpose, it is computed the average share of electric consumption during ToU hours. The number of ToU hours offered a day is 9 hours[7]. This means that ToU hours span $\frac{9}{24} = 37.2\%$ time in a day: so if a house has storing capacity, the electric consumption share at night should be higher than this value. A more conservative value of 45% of consumption during ToU hours is set as the minimum share to determine if a house owns a TES system. Four groups are clustered:

- Average share between 45-60%, into *storeh_45_60*.
- Average share between 60-75%, into *storeh_60_75*.
- Average share between 75-100%, into *storeh_75_100*.
- Average share under 45% in ToU: no storage capacity.

This selection does not estimate a capacity value, but enhances the program functioning by reducing the number of clients in every calculation.

Storing Capacity Selection Tool. Critical Day and Limit Temperature.

The aim of this tool is to find the *Critical Day*. The *Critical Day* is defined as "the day on which the boiler used the maximum number of consecutive loading hours ($n_{l.h.}$) historically". A house with a ToU tariff exhibits a low electric demand during daytime, to later increase its consumption level at night: this notorious increase of consumption is mainly caused by the boiler load.

A limit ToU-load pattern is performed for every client by studying his average consumption behaviour.[8] This pattern sets two consumption limits, a lower one for daytime and an upper one for night time, both created from the record. The lower limit bounds those day when there is no

1. Lower bound: 1.1 times the average hourly consumption in daytime along winter time. Increasing this value 1.1 times is intended to cover also higher daily consumptions caused by the possible use of other electric devices. It is set from 7 am to 10 pm.
2. Upper bound: maximum average consumption metered during winter nights, since there is no higher average consumption recorded historically. It is set from 10 pm to 7 am of the next day.

The storing capacity selector works with winter consumptions. Only winter time is studied because it can be ensured that, in a house located in Finland, most of the energy is used for heating purposes during this period of the year. Also because when running the program, some houses show higher consumption during summer nights than during winter nights, being in disagreement with the kind of customer of this study. From both bounds a daily *Consumption Limit* is also established. This limit represents the amount of electricity this house would consume before the storage is depleted.

$$Consumption\ Limit = Upper\_bound \cdot 9 + Lower\_bound \cdot (22 - 7)\ ^{9} \qquad (2)$$

[7] From 10 pm to 7 am on the next day.
[8] Program´s body in *Day Selector*, Annex 3.
[9] The upper bound is multiplied by the 9 hours of ToU (from 10 pm to 7 am) and the lower bound is multiplied by the hours out of ToU (from 7 am to 22 pm).

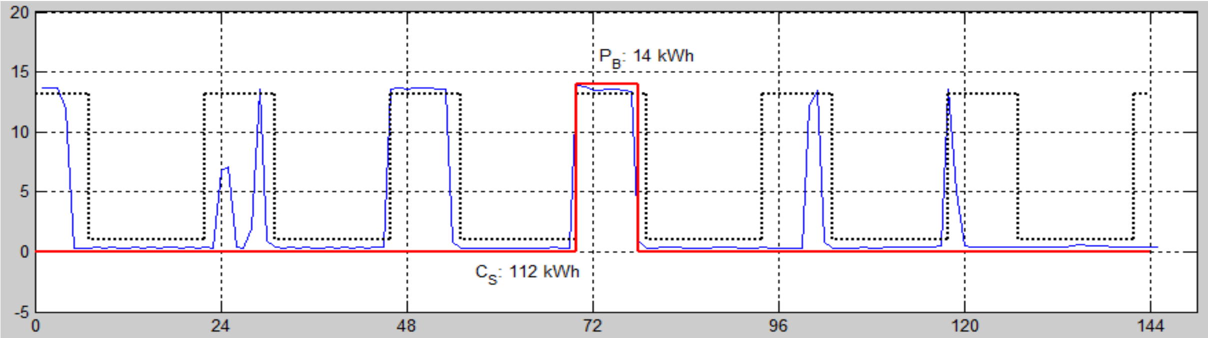


Figure 4 - In black, the pattern; in blue, the real profile; in red, the Capacity selected. Over the selected day, the computed Storage Capacity and Boiler Power.

The *Storing Capacity Selector Tool* contrasts the created pattern with the metered load profile. During several days of the year the house exhibits a similar behaviour to this limit pattern, meaning that there is no extra consumption because the storage is not depleted. Between all these days, there is one when the average temperature outdoors is minimum: this is selected as the *Day Limit*, and this minimum average temperature is selected as the *Temperature Limit*.

If the temperature surpasses this limit, historically they appear unexpected consumption peaks during daytime: this suggests that the storage is depleted. In order to guarantee the selection of a day that accomplishes the specifications imposed by the pattern, the program follows an iterative adjustment of the bounds and *Consumption Limit*. This adjustment is done because consumption trends vary dramatically from one household to another.[10] The specifications inside the program are the following:

- Daily consumption must be lower than Consumption Limit imposed. This value does not change with the iterations.
- Consumption during daytime must be lower than the lower bound, meaning there was no extra heat consumption out of ToU. Neither this value is adjusted with consecutive iterations.
- Consumption percentage during ToU ought to be higher than 90% times the upper bound. This specification ensures the storage was filled-in from very low levels of storage to its total capacity. In every iteration this percentage is decreased by 5%.
- The upper bound is adjusted by multiplying it with a correction factor of 0.95 in every iteration.
- *Temperature Limit* must be lower than the reference one. Every time a day accomplishes all these specifications, the *Temperature Limit* is recalculated as the minimum between the previous limit and the temperature of the studied day.

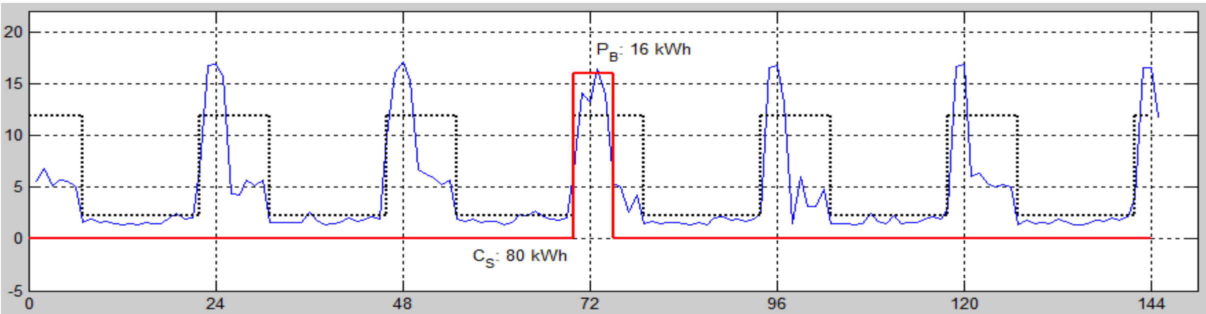


Figure 5 – In this case the number of hours is shorter, but P_B is measured as 16 kW (high value).

[10] See *Limitations of the estimated capacity* on the next page.

Limitations of the estimated Storing Capacity and Boiler Power

It is important to highlight the limitations of these estimations. With the available data it is almost impossible to prove the accuracy of C_S and P_B; but it can be ensured that historically, when the needed heating consumption overcomes the value of stored value of C_S it usually appears a daytime consumption higher than the average. This extra consumption out of *ToU* hours means a lack of energy in the balance during cold days, reason why it is understood as a depletion of the energy stored.

It is also important to mention that each customer's routine and different storage size vary dramatically from one to another, reason why the pattern used to search for the *Critical Day* and *Limit Temperature* is adjusted iteratively inside the program. This is an empirical solution given after studying different demand profiles.[11] Thanks to this iterative adjustment it is found a *Critical Day* from which estimate C_S and P_B for each house.

When the record shows a constant high consumption level during *ToU* hours the P_B estimation fits this value; but in accordance with other consumption profiles registered it can be seen that not all the houses load with a constant power level during *ToU* hours[12]. The appearance of an irregular load profile at night can be mainly motivated by three situations:

1. The boiler follows a different control pattern varying its power level.
2. Other electric devices or appliances are also loading during ToU hours.
3. This customer does not own a TES system and another appliance is exhibiting this high consumption.

In case of matching the first or the second situation, the estimated P_B is computed as the integer value of night peak consumption, thus introducing an error since this value is incorrectly estimated. The third situation leads to the most dramatic case, since it means the program has detected a TES consumption pattern incorrectly. But if this situation is reached for example, in *Figure 2*, it means there is another appliance that consumes more than the 16 kW per hour every night. This value is too high to be any other common appliance in a house. [31]

It would be interesting to contrast the accuracy of this tool to compute real values, but the customer reserves the right not to tell the details of the system installed in his house; even the building characteristics affect the consumption dramatically. The simplicity of this cluster tool allows reaching a generic storing capacity independently from the actual TES system of the house, and a tangible value to perform the simulation and demand control. Some examples of the *Storage Tank Capacity Selector Tool*'s output can be found in *Annex 13*.

[11] Annex 2, *LEGO function*.
[12] See *Figure 4* on the previous page

Mathematical model of a TES system

The data to perform the storage level simulation are the estimated *Boiler Power* ($P_B, in\ kW_{elec}$) and the *Storage Tank Capacity* ($C_S, in\ kWh_{heat}$). Several mathematical models of a TES system have been presented in different literature reviews [17] [18] [19] [20] by addressing the problem from the customer's point of view: complex models of tank level simulation are developed when, in addition to the P_B and C_s, also the tank's dimensions, storage medium and other relevant values from the system are known beforehand. Since these data are not available in this study, the selected mathematical model performs in a simple way the energy balance that occurs inside the storage tank [15] regardless the system specifications:

The parameter S_i (kWh_{heat}) represents the storage level at i hour

$$Hourly\ balance \qquad S_{i+1} = S_i + P_B \cdot x_i - w_i \qquad i \in \mathbb{N}(1..24), \forall S_j \ \ 0 \leq S_j \leq C_s \tag{3}$$

- S_i (kWh_{heat}) is the storage level at the beginning of that hour
- $S_{i+1}(kWh_{heat})$ is the initial storage level on the following hour
- $P_B(kW_{heat})$ is the *Boiler Power*
- $x_i(h) \in \mathbb{N}\{0..1\}$, being 0 when the boiler is *off* and 1 when the boiler is *loading*.
- $w_i(kWh_{heat})$ is the hourly heating demand

Two possible situations can arise:

$$1)\ Charging\ mode \qquad S_{i+1} = S_i + P_B - w_i \qquad [kWh_{heat}] \tag{4}$$

The boiler is loading from grid, and its energy loaded in one hour ($P_B \cdot 1h$) is partly used to charge the tank and partly used to cover the heat demand of that loading hour.

$$2)\ Discharging\ mode \qquad S_{i+1} = S_i - w_i \qquad [kWh_{heat}] \tag{5}$$

The boiler is off, and the heating demand is fully covered by the heat reservoirs inside the storage tank.

The two different functioning modes of a TES system are represented through both equations; the only missing parameter to complete the formulae is the *hourly heating consumption* (w_i), which is introduced on the following chapter.

Chapter 3. MODEL OF DAILY HEATING CONSUMPTION

Introduction

A linear regression model is studied in order to find a correlation between hourly heating use (w_i) and temperature outdoors (T). The assumption is as follows, "the energy used for space heating is straight related to the average temperature outdoors of the day".

Direct heating systems are usually controlled by a thermostat, which switches the boiler when the temperature indoors varies from the control one: electricity is straight converted into heat and used for heating purposes, exhibiting both profiles similar trends. This suggests a regression model between real electric consumption and temperature outdoors.

On the other hand, this assumption cannot be applied to houses with heat storage tank. TES systems generate a mismatch between the metered electric consumption and the real heating use profile of the house. From metered data it is shown the total electricity consumption, so distribution companies cannot see "downstream the tank" this heating consumption curve. *"TES systems are widely recognized for decoupling electricity production and demand"* [7].

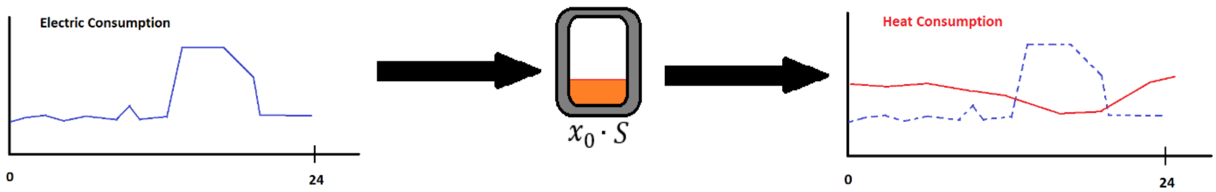


Figure 6 - On the left in blue, metered consumption upstream the tank; on the right, heat consumption in red. Example of the decoupling between the two profiles.

Formulae

By applying energy balance, the functioning of these systems can be shown and the approach to the regression model can be set. In order to perform the simulation downstream the tank, some assumptions are followed:

- C_S and P_B are correctly estimated.
- Based on the C_S estimation, two consecutive days are selected. During these days the rate of ToU consumption is almost the full capacity of the storage tank[13]. There are no consumption peaks during daytime.[14]
- The boiler only loads during ToU at full capacity, exhibiting an almost constant demand.
- The boiler switches off during ToU when C_S is reached and the storage is full.
- The average heat consumption per hour ($\overline{w}$) extracted from the tank is expressed as function of the temperature: the colder the temperature, the higher amount of heat is needed per hour and vice versa.

$$\overline{w_{day}} = \sum_{i=1}^{24} w_i = f\left(\overline{T_{day}}\right) \qquad \overline{T_{day}} = \text{average temperature of "day"} \qquad (6)$$

[13] These rates can be appreciated on the following page. They are 0.9519 and 1.

[14] Picture 1, next page.

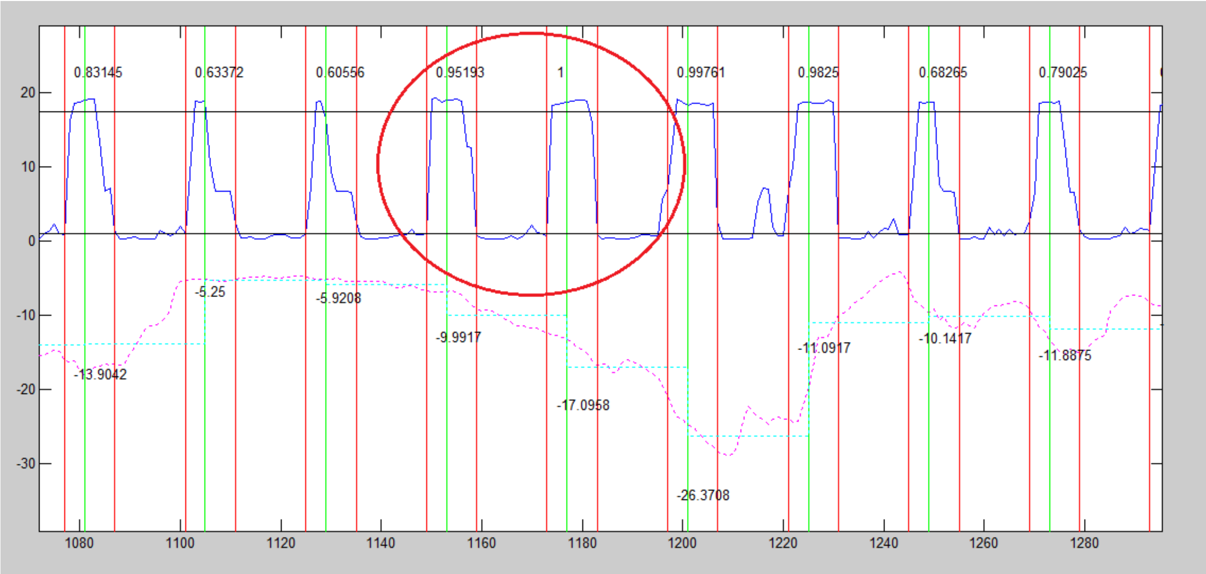


Figure 7 - Simulation using a clustered house (storeh_75_100), Central Finland

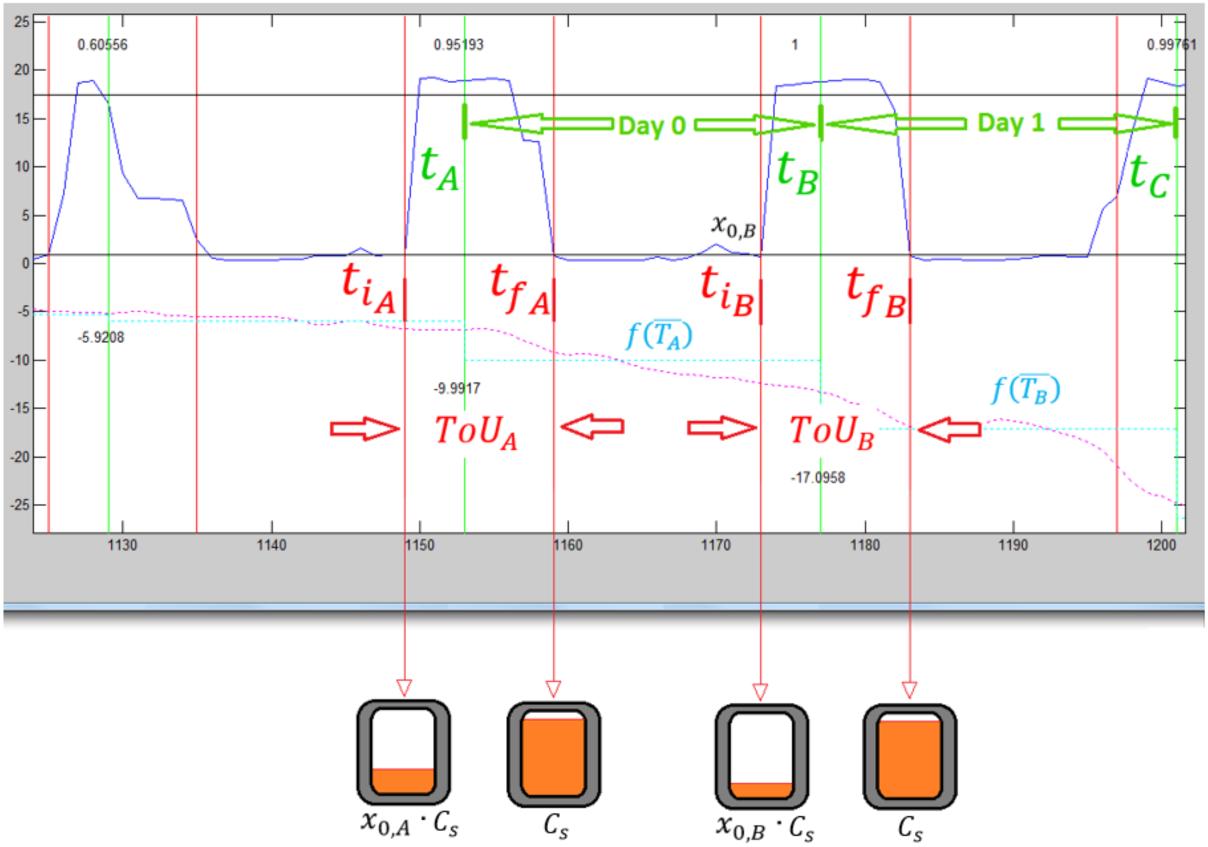


Figure 8 - Same house. Rendering of tank´s levels at different hours.

In these figures it can be observed how the consumption falls dramatically at the end of ToU (t_{f_A} and t_{f_B} in Picture 2): this is understood as “the tank is full”. There are not dramatic peaks of consumption registered during daytime. If the storage tank is full, the thermostat gives the order to switch off the boiler.

Both days the storage is filled in until reaching C_s since there is no bigger night time consumption historically metered. Other variables in use:

- $t_{i_{day}}$ referred to the initial ToU hour and $t_{f_{day}}$ final ToU hour.[15]
- t_A, t_B and t_C referred to 12 am of *day A*, *day B* and *day C*, so $(t_A - t_B) = 24$ hours.
- $ToU_{day} = t_{f_{day}} - t_{i_{day}}$ to refer to each ToU period used to load the tank for the next day. The length of ToU is constant and equal to 9 hours during night time.
- $x_{o,day} \in (0..1)$ represents the storage tank's filling percentage.[16]

Energy balance

$$Daily\ heat\ consumption \begin{cases} W_A = (t_B - t_A) \cdot \overline{W_A} = (t_B - t_A) \cdot f(T_A) = 24 \cdot f(\overline{T_A}) & day\ A \\ W_B = (t_C - t_B) \cdot \overline{W_B} = (t_C - t_B) \cdot f(T_B) = 24 \cdot f(\overline{T_B}) & day\ B \end{cases} \quad (7)$$

The real heat consumption of the house varies hourly with the temperature profile, but it can be simplified by using the average temperature along 24 hours every day. [17]

$$Energy\ Eq. in\ the\ tank: \qquad (ToU_B \cdot P_B) + x_{0,B} \cdot C_s - ToU_B \cdot f(\overline{T_B}) = C_s \quad (8)$$

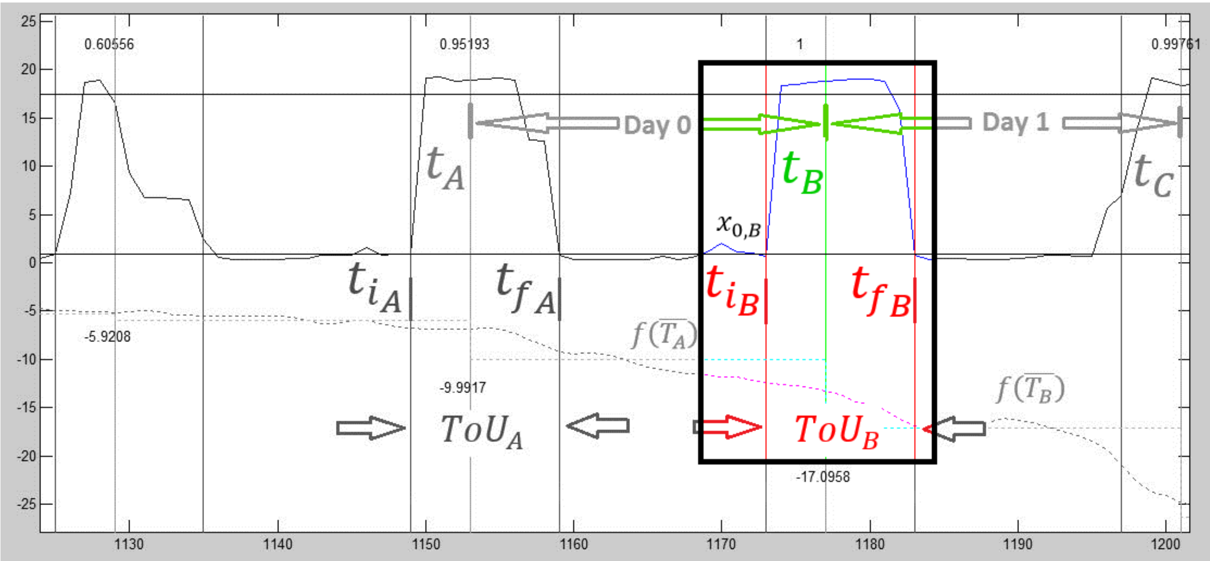


Figure 9 - Energy balance during ToU_B. "Day 0" on the graph is "day A"; "Day 1" is "day B"

This equation performs the energy balance during ToU_B: the tank is filled again until C_s; $x_{0,B} \in (0..1)$ so $x_{0,B} \cdot C_s$ expresses the remaining level inside the storage tank at 10 pm of *day A* when $\mathrm{ToU_B}$ hours start.

$(ToU_B \cdot P_B)$ is "boiler energy load during ToU_B", expressing the energy loaded to fill the tank again until C_S; by using the average temperature's simplification, $ToU_B \cdot f(\overline{T_B})$ expresses "real heat consumption of the house" during loading hours.

[15] For this company in particular t_{o_i} is 10 pm and t_{f_i} is 7 am, so $ToU \in (22{:}00 - 7{:}00)$

[16] Do not confuse this vale with x_i used in the *Mathematical Model*.

[17] In this estimation is has been considered that during ToU_B the average temperature is $\overline{T_B}$. The error committed is not representative since 7 hours towards 9 of ToU belong to *day B* in this range, and only 2 hours to *day A*.

$$Eq. from\ 7\ am\ (A)\ to\ 10\ pm\ (A): \quad C_s - x_{0,B} \cdot C_s = \left(t_{i_B} - t_{f_A}\right) \cdot \overline{w_A} = (t_{i_B} - t_{f_A}) \cdot f(\overline{T_A}) \qquad (9)$$

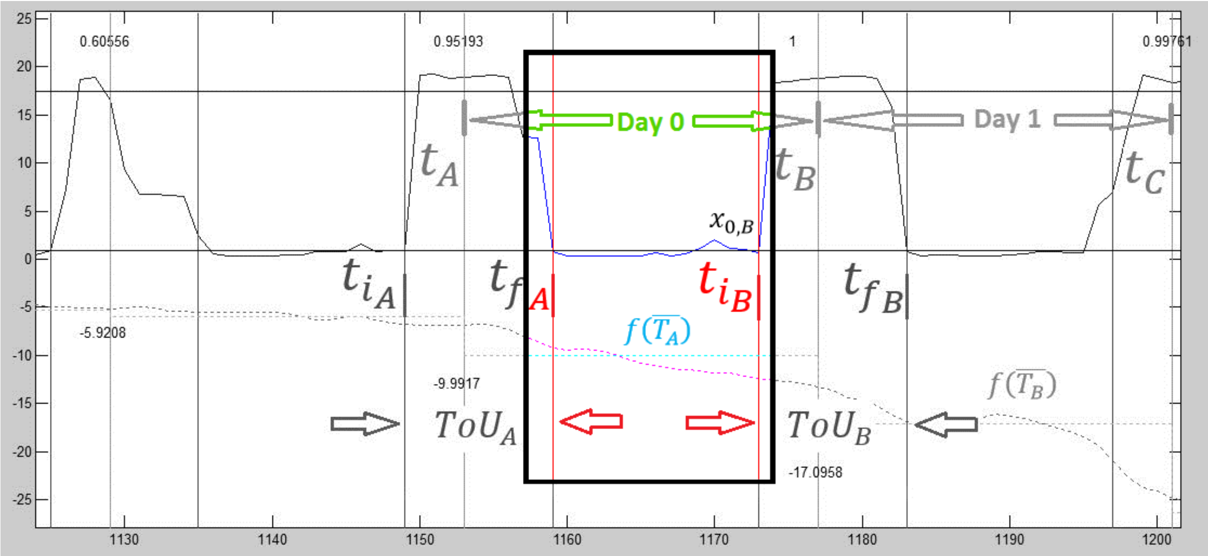


Figure 10 - Daytime balance during Day A

When ToU_A is finished at t_{f_A}, all the heating demand of *day 0* is supplied by the tank. The energy extracted from the tank during *day A* to cover the daytime demand is $C_s - x_{0,B} \cdot C_s$

The remaining part inside the tank at t_{i_B} $(x_{0,B} \cdot C_s)$ plus energy bought at ToU_B gives C_s again (at t_{f_B})[18]. There's still some remaining level in the tank that was not used yesterday, during *day A* the consumption from the tank was:

$$From\ (9)\ and\ (10) \qquad (ToU_B \cdot P_B) = ToU_B \cdot f(\overline{T_B}) + \left(t_{i_B} - t_{f_A}\right) \cdot f(\overline{T_A}) \qquad [kWh_{heat}] \qquad (10)$$

$$Electricity - Heat\ conversion: \qquad W_B = COP_b \cdot (ToU_B \cdot P_B) \qquad [kWh_{heat}] \qquad (11)$$

W_B is "total heat input during ToU_B.[19]

$$And\ the\ time\ equality: \qquad t_{i_B} - t_{f_A} = 24 - ToU_B \qquad [h] \qquad (12)$$

Combining (11), (12) and (13) the searched formula is reached: the relationship between heat consumption and temperature outdoors.

$$W_B = COP_b \cdot (ToU_B \cdot P_B) = COP_b \cdot \{ToU_B \cdot f(\overline{T_B}) + (24 - ToU_B) \cdot f(\overline{T_A})\} \qquad [kWh_{heat}] \qquad (13)$$

Grouping and generalizing,

$$W_B = 24 \cdot COP_b \cdot f(\overline{T_A}) + ToU_B \cdot COP_b \cdot [f(\overline{T_B}) - f(\overline{T_A})] \qquad [kWh_{heat}] \qquad (14)$$

This formula shows how the regression model must be built: the determinant temperature for filling in the tank is not today's temperature but yesterday's one.

[18] Consult *Picture 2*
[19] Assuming a real boiler, with $COP \in (0.88, 0.99)$

Regression Model

The statement from formula: the amount of heat loaded today during ToU hours to fill-in the storage tank until its 100% capacity is linearly related to yesterday's outdoors temperature, with a deviation caused by the need of heating supply while charging the tank. From (15):

$$W_B = 24 \cdot COP_b \cdot f(\overline{T_A}) + ToU_B \cdot COP_P \cdot [f(\overline{T_B}) - f(\overline{T_A})] \qquad [kWh_{heat}] \qquad (15)$$

And deriving this formula into general indexes, the linear model is obtained:

$$W_i = f(T_{i-1}) + deviation \qquad [kWh_{heat}] \qquad (16)$$

This formula sets the regression model, from which the linear part as well as the deviations are studied in order to reach an accurate model of consumption.

The ToU consumption is computed and plotted with the temperature of the previous day in Matlab®.[20] This figure shows a single measurement of heat consumption during ToU hours per day and its relationship with the average temperature of the day before:

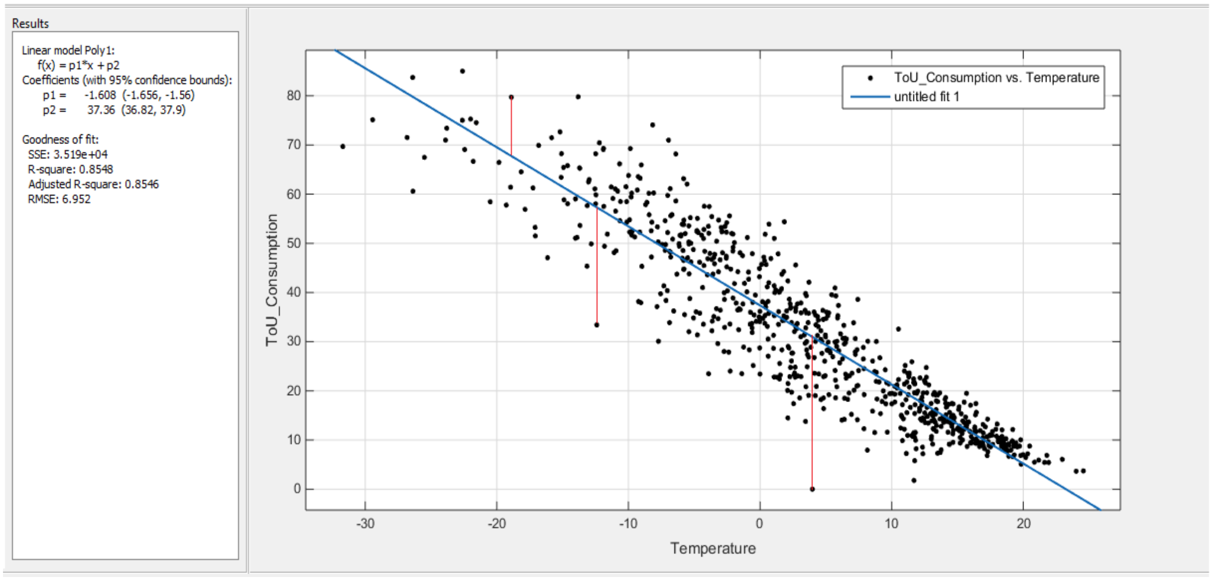


Figure 11 - Regression model of real metered data from House_47, Jyväskylä. Some large deviations shown in red.

The linear regression reaches acceptable R-squared values (this sample house in particular has a "goodness of fit" of 85,45%), and deviations can be clearly seen (some of them in red): the linear model is accepted.

The aim of this linear regression model is to simulate in a simple way the hourly heat consumption $w_i(T)$ to start the control simulation. Regression is made out of night consumptions (ToU hours) because it can be ensured the consumption during these hours is mainly used by the TES system. Houses with small storing capacity typically show high consumptions during daytime (out of ToU). This daytime extra-consumption of heating is not contemplated in this model since only ToU consumptions are integrated: reason why this linear model is corrected on the following section.

[20] *Model of Daily Consumption*, Annex 4

Adjustment of the Linear Regression model to meet the scale of demand

An *Average house* in Jyväskylä is performed as an example where the linear model does not cover all the demand. It is computed from households clustered into *storeh_60_75* inside program's body.

```
Command Window
>> Simulation
group of study: storeh_60_75
number of house (0 if AVERAGE HOUSE): 0
_____________________________________________________
                 ANALYSIS RESULTS
_____________________________________________________
Maximum Capacity of the House: 89.9928
Consumption Limit of Discrimination: 127.6516
critical day: 13-Mar-2013
critical day is a WINTER DAY
Percentaje out of max capacity: 0.83185
Percentaje Stored of critial day: 0.72196
Day Temperature when Limit of Storage Happens: -16.81
Temperature of previous day (that leaves the Storage
Day: 438
Hour: 10489
_____________________________________________________
             SUMMARY OF THE HOUSE
-----------------------------------------------------
Number of House: 1
TOTAL CONSUMPTION: 45004.1055 kWh
CAPACITY: 89.9928 KWh
TEMPERATURE: -14.5054 °C
_____________________________________________________
```

Figure 12 - Sample of the program output

This house has an electric consumption of 45 MWh_{elec} during the metered time span (2 years), which means an annual heat consumption of 22,5 $\frac{MWh_{elec}}{year}$. Compared to statistics [33], this can be the consumption of a small a house where two people live in Finland:

Table 1- Consumption per capita in different countries

country	annual electricty consumption per capita	
Iceland	25,127 kWh	
Norway	24,861 kWh	
Finland	15,812 kWh	
Sweden	15,679 kWh	
Canada	15,666 kWh	

Based on statistics, around the **84%** of the total electric consumption of a common household is used for heating purposes in Finland:

Table 2 - Consumption share in Finland [4]

Consumption (GWh)	2008	2009	2010	2011
Heating of residential buildings	51,241	54,872	60,589	51,863
Housing, total	61,051	64,905	70,878	61,884
% Heating out of Total	0,84	0,85	0,85	0,84

The same rate is applied to this house to compute the expected heating consumption:

$$Annual\ Heat = 0{,}84 \cdot \frac{MWh_{heat}}{MWh_{elec}} \cdot 45\ MWh_{elec} = \mathbf{37{,}8\ MWh_{heat}} \tag{17}$$

And, by using the performed linear model from ToU hours, this value is also computed:

$$Annual\ Heat\ (ToU) = \mathbf{30{,}480\ MWh_{heat}} \tag{18}$$

Result: this house has consumed 30,48 MWh_{heat} in ToU hours to charge the storage. This value is far from the theoretical total consumption (37,8 MWh_{heat}), meaning there is an expected amount of heat of $37{,}8 - 30{,}48 = \mathbf{7{,}32\ MWh_{heat}}$ that cannot be supplied by the tank by only using ToU hours; this energy difference is loaded at any time whenever the customer demands it, regardless the price evolution of the day. This extra consumption out of ToU is not covered by the linear model, and also needs to be contemplated.

The linear model needs to be scaled up according to the estimation of annual heating demand. It is automatically adjusted by the program[21]:

$$Hourly\ model\ (from\ ToU): \qquad w_i = \frac{W_{day}}{24} = -0.08293 \cdot \overline{T_{i-1}} + 2.061 + deviation \tag{19}$$

$$Correction \quad \widehat{w_i} = w_i \cdot \frac{Heat\ consumption\ \left(\frac{kWh_{elec}}{year}\right)}{ToU\ consumption\ \left(\frac{kWh_{elec}}{year}\right)} = w_i \cdot \frac{0.84 \cdot Annual\ kWh_{elec}}{ToU\ \ kWh_{elec}} \tag{20}$$

$$Corrected\ Model\text{:}\ \widehat{w_i} = w_i \cdot \frac{37.8 \frac{kWh_{heat}}{year}}{30.48 \frac{kWh_{heat}}{year}} = -0.1028 \cdot \overline{T_{i-1}} + 2.556 + deviation' \tag{21}$$

[21] *Model of Daily Consumption*, Annex 4. The following models are automatically computed inside the program.

Once the dimension fitting is done, the integral value obtained from the estimation is 38,05 MWh_{heat} (*figure 12*) of the total consumption of 45 MWh_{elec} (84,6% share). This adjustment helps the algorithm to use a more feasible consumption level of each customer.

```
Command Window
>> Model_Daily_Cons
group of study: storeh_60_75
number of house (0 if AVERAGE HOUSE): 0
>> sum(Estimation)

ans =

   3.8051e+04

>>
```

Figure 13- Validation of the estimation

The correction is automatically performed inside the program and shown on screen:

```
Command Window
Area of study? JYVÄSKYLÄ=1; HELSINKI=2; TAMPERE=3; TAVASTIA=4: 1
group of study: storeh_60_75
number of house (0 if AVERAGE HOUSE): 0
----------------------------------------------------
 Linear model: q(T)= -0.10286*T+2.5565
 Residuals: Normality test passed
----------------------------------------------------
_____________________________________________________________________
                 ANALYSIS RESULTS
_____________________________________________________________________
Maximum Capacity of the House: 89.9928
Consumption Limit of Discrimination: 127.6516
critical day: 13-Mar-2013
critical day is a WINTER DAY
Percentaje out of max capacity: 0.83185
Percentaje Stored of critial day: 0.72196
Day Temperature when Limit of Storage Happens: -16.8167
Temperature of previous day (that leaves the Storage empty): -9.1125
Day: 438
Hour: 10489
__________________________________________________________________
             SUMMARY OF THE HOUSE
------------------------------------------------------------------
Number of House: 1
TOTAL CONSUMPTION: 45004.1055 kWh
CAPACITY: 89.9928 KWh
TEMPERATURE: -14.5054 °C
fx __________________________________________________________________
```

Figure 14 - Sample of the program output

Deriving this into general terms:

$$Linear\ Model = Regression\ model\ (ToU\ hours\ of\ metered\ data)\ \&\ Size\ correction$$

Once the linear model is settled, only the study of the deviations is missing to complete the consumption model; this error analysis is later performed at Chapter 4.

Chapter 4. GENETIC ALGORITHM

History

The Theory of Evolution, postulated by the English naturalist Charles Darwin in 1859 in his book “The origin of species”, presents what is known as *Natural Selection* as an explanation to the great variety of species and subspecies that populate the Earth. After all his life observing and comparing different species in different environments, Darwin got to the conclusion that species, in order to survive on their environment, follow a natural selection where best adapted specimens of a population survive to climate specifications and variations and reproduce themselves on future generations, whereas worst adapted individuals tend to disappear and not to reproduce. Darwin got the idea, but without theoretical scientific fundament; it has been tens of years later, when after combining his theory with other scientists’ theories such as Mendel’s laws of inheritance (published in 1865 but ignored until the beginning XX Century) [34], and all discoveries made thanks to the Human Genome studies when the Genetic Theory was born.

“*In the 50s and the 60s several computer scientists independently studied evolutionary systems with the idea that evolution could be used as an optimization tool for engineering problems. The idea in all these systems was to evolve a population of candidate solutions to a given problem, using operators inspired by natural genetic variation and natural selection.”* [35]. It was John Holland, who during the 60’s, developed what are known as Genetic Algorithms (commonly referred as GA in literature reviews). These algorithms don’t have a unique solution, same as in nature where several individuals with different characteristics are adapted and coexist in the same environment. The survival of an individual is guaranteed when this individual owns the characteristics needed to fit the specifications demanded by the environment. The presence of individuals with different characteristics but all of them valid to survive is what gives a high rate of variety to a population, reaching different valid solutions in the evolution equation.

“There is no rigorous definition of "Genetic Algorithm" accepted by all the evolutionary−computation community that differentiates GAs from other evolutionary computation methods. However, it can be said that most methods called "GAs" have at least the following elements in common: populations of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring.” [35]

“The chromosomes in a GA population typically take the form of bit strings. Each locus in the chromosome has two possible alleles: 0 and 1. Each chromosome can be thought of as a point in the search space of candidate solutions. The GA processes populations of chromosomes, successively replacing one such population with another. The GA most often requires a fitness function that assigns a score (fitness) to each chromosome in the current population. The fitness of a chromosome depends on how well that chromosome solves the problem at hand.” [35]

Application to this case.

In this study, the individuals are 24-component vectors (24 chromosomes), same as the number of readings given by the meter in one day. Each allele has two possibilities: “0” allele means “discharging mode of the TES”, whereas “1” allele means “charging mode of the TES”.

The creation steps of new generations are presented with two sample individuals:

Reproduction: *this operator selects chromosomes from population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce.* [35]

parent 1	1	1	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0
parent 2	1	1	1	0	0	1	0	1	0	0	0	1	0	1	1	0	0	0	1	1	1	0	1	1

Crossover: *this operator chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring* [35]. An example of crossover of 0,5:

parent 1	1	1	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0
parent 2	1	1	1	0	0	1	0	1	0	0	0	1	0	1	1	0	0	0	1	1	1	0	1	1
						↓													↓					
offspring 1	1	1	0	0	1	0	1	1	0	0	1	0	0	1	1	0	0	0	1	1	1	0	1	1
offspring 2	1	1	1	0	0	1	0	1	0	0	0	1	1	1	0	0	1	0	1	1	0	0	1	0

Mutation: *this operator randomly flips some of the bits in a chromosome*

child 1	1	1	0	0	1	0	1	1	0	0	1	0	0	1	1	0	0	0	0	1	1	0	1	1
child 2	1	1	1	0	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1	0	0	1	0

One common application of GAs is function optimization, where the goal is to find a set of parameter values that maximize a complex multiparameter function [35]. The goal of this study is to create daily control vector (chromosome) to give to the customer as a pattern to switch the boiler. In this way, the heating demand can be shifted from peak hours.

The *fitness function* on which this control is based is as follows:

Offer cheap hours to load the boiler in order to minimize the costs of electricity

Where several physical restrictions[22] must be accomplished:

- Storage tank level must remain below the 100% of its capacity every hour
- At the end of the day the level must be within the range from 45 to 65% of capacity.
- The number of loading hours must tend to be the minimum needed to supply the demand

[22] Annex 6, *Internal Constrains inside GA*.

The following block diagram shows all the steps of the standard Genetic Algorithm:

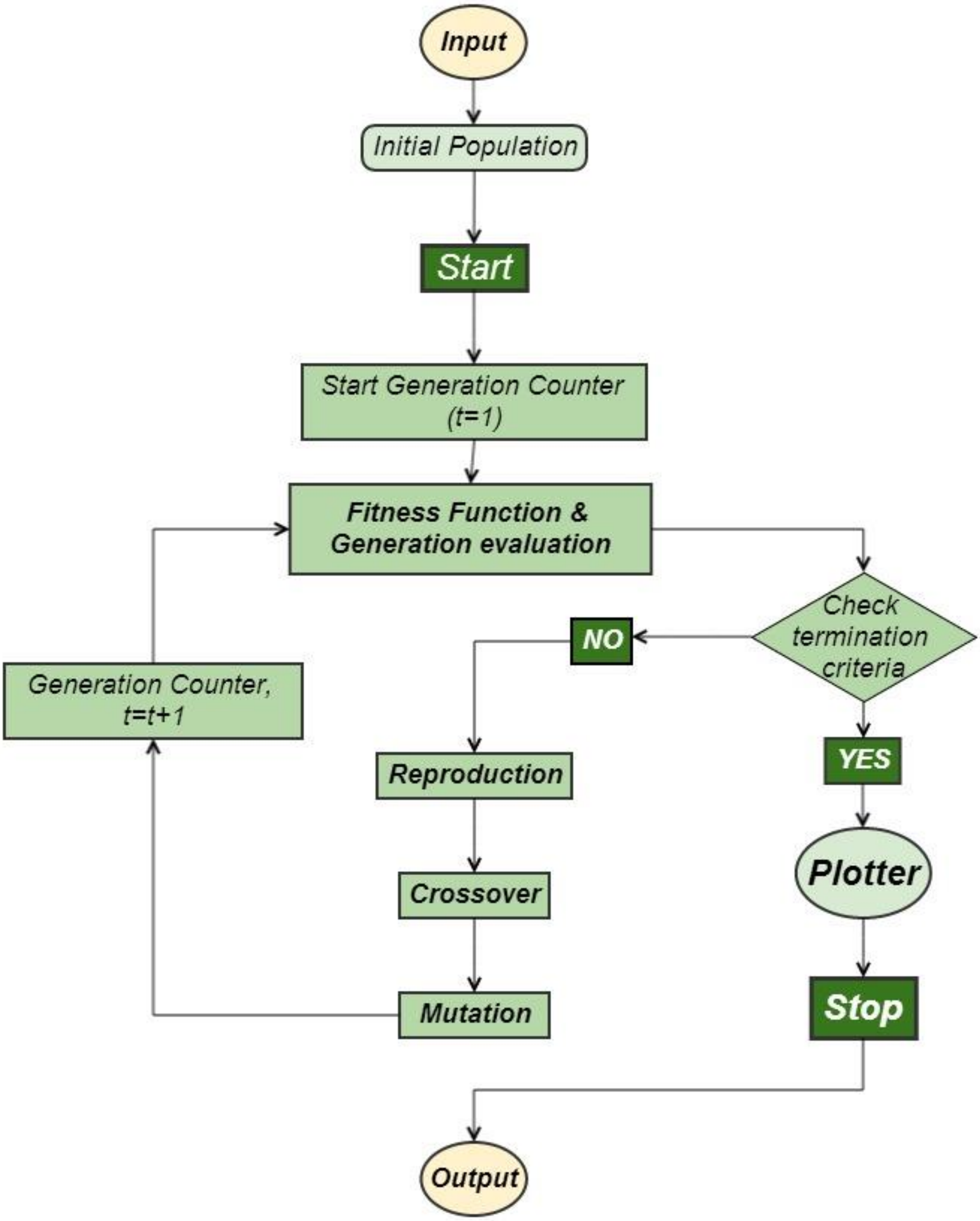


Figure 15 - Block diagram of the standard Genetic Algorithm

This algorithm is already developed inside the *Optimization Toolbox* of Matlab®. The controller sets all the specifications of the optimization, and Matlab® operates it.[23]

[23] Specifications in Annexes 6 and 7; GA call inside Annex 5, *Stochastic Genetic Algorithm*, GA in Matlab®.

Control vector $\bar{x}$

$$\bar{x} = (x_1, x_2 \dots x_n) \quad where\ \bar{x} \in \mathbb{R}^{24}, \qquad where\ \forall i \quad x_i \in \mathbb{N}^*(0,1) \tag{22}$$

Where "$x_i = 0$" means "don't load in this hour" and "$x_i = 1$" means "load in this hour"

Fitness function

$$GA\ function: \quad f_{fitness} = Min \sum_{i=1}^{h} (P_B \cdot x_i) \cdot K_{Elspot,i} \qquad [€] \tag{23}$$

Where $K_{Elspot,i}$ $(€/kWh)$ represents the Elspot price at i hour. This is the equation the GA has to satisfy daily to minimize the costs of electricity. [24]

Constrains:

i. Minimum heating requirement

$$\textstyle\sum_{i=1}^{h} x_i \cdot P_B = n_{l.h.} \cdot P_B \geq W - S_0 \ \rightarrow\ \sum_{i=1}^{h} x_i = n_{l.h.} \geq \frac{\mathrm{W}-S_0}{P_B} \tag{24}$$

- W (kWh): amount of heat needed during that day [25]
- S_0 (kWh): initial storage level of that day $(t = 0, t \in [0{:}\,24]\ hours)$
- $n_{l.h.}$: number of charging hours $\quad n \in \mathbb{N}^*(0{,}24)$
  - *Restriction:* "the amount of hours the storage will load today has to be at least the minimum number of hours required by the system based on the consumption forecast, so costumer's demand is satisfied". Basically, "x" will tell the amount of "1" in our vector.

ii. Net storage capacity limit imposed by constraint every hour[26]

$$S_{i+1} = S_i + P_{max} \cdot x_i - w_i < C_s \qquad [kWh_{heat}] \tag{25}$$

- *Restriction:* The maximum storage capacity cannot be overcome at any time.

iii. The storage level should remain between the bounds

$$0 \leq S_i \leq C_S \qquad [kWh_{heat}] \tag{26}$$

[24] Annex 5, *Fitness function inside GA*.
[25] Its derivation is later explained at Chapter 5, *Stochastic Genetic Algorithm*.
[26] This equation was already presented in the Mathematical Model.

Chapter 5. STOCHASTIC GENETIC ALGORITHM

Genetic Algorithm is widely spread used to find an optimal solution of a given problem among the huge existing number of possibilities for solutions. Optimal solutions are reached by applying Natural Selection's steps, such as crossover, mutation and inversion: starting from an existing population, this algorithm recombines and modifies solutions until reaching an optimal one which satisfies all the constraints.

Stochastic Genetic Algorithm (SGA) is the given name to the GA's application, when even the first population has been created with pre-selected criteria, altered by the controller. It has been developed to overcome the low global convergence speed of the standard Genetic Algorithm and to keep optimal solutions within an already accepted shape.

It is named "stochastic" because it simulates the stochastic behaviour of electric demand: historically, the metered levels of consumption present deviations from the linear model. These deviations have certain probabilities of happening along the year: when it is understood how consumption deviations are distributed in the time span, these deviations can be afterwards modelled and used as parameters in consumption forecast. The use of probabilities gives a controlled randomness to the performed forecasting process in this work, and simulates the stochastic variations of electric demand in a more realistic way.

By applying regression to the electric record, a linear model of consumption is generated for every house. This model consists of a linear part (function of the temperature) and a certain deviation. Inside SGA these historical deviations are studied and clustered into different levels, and are afterwards simulated.

When doing distribution analysis it is found that every deviation from linearity, historically, has a probability to appear. This means the consumption of the house has typically varied from one of these levels to another, but always inside consumption bounds. All this information is recorded inside SGA in order to generate the primitive population of vectors.

Every response vector inside this initial population is a feasible control to the house since they have been generated from its historical record. Every deviated consumption level has its own response vector assigned, meaning that each response vector is likely to appear according to the probability of its corresponding deviation level: SGA translates each probability into a proportional number of copies of each individual inside the first population of response vectors.

The improvement of the standard GA starts by generating this initial population of feasible solutions, from whom an optimal offspring-solution evolves. In this evolution process, when all the initial individuals present the same chromosome in the same position of the string, this chromosome is selected as valid. This chromosome is accepted and automatically fixed; in this way the GA does not modify it, and restricts the evolution of the optimal solution only inside the primitive population. This enhances the convergence speed of the standard GA by reducing notoriously the number of combinations.

The best adapted response vector is determined, then its goodness is studied, and finally is given as a control to the customer.

The working procedure of the SGA explained in its principal steps:

1. **Generating the Initial Population**

   Same as in nature, individuals with different characteristics have different probabilities of appearing: understanding and modelling these probabilities for different groups is the starting point of this study.

   a) Hourly heating demand from linear regression model
   b) Deviations: errors study
   c) Error levels simulation
   d) Combination of simulated errors and linear model.
   e) First and Second-Generation response vectors
   f) Creation of Initial Population

2. **Standard GA specifications & Convergence speed improvement**

   Thus, optimal solutions are reached.

   a) Characteristics used inside GA
   b) Convergence speed improvement
   c) Temperature Forecast: ARMA Model

3. **Output of the SGA**

   Input options given by the program and result.

   a) Calculations and Results
   b) Shape of the response vector

A sample house is selected in order to explain in detail all the steps: *house 47 in Jyväskylä*.

This house is clustered into *storeh_60_75*. [27] It has been selected because it shows a typical consumption. From now it will be referred as ***House 47*** in the analysis.

[27] Annex 1, *Cluster Tool*.

1. Generating the Initial Population:

a) Hourly heating demand from linear regression model

Running the program for House 47:

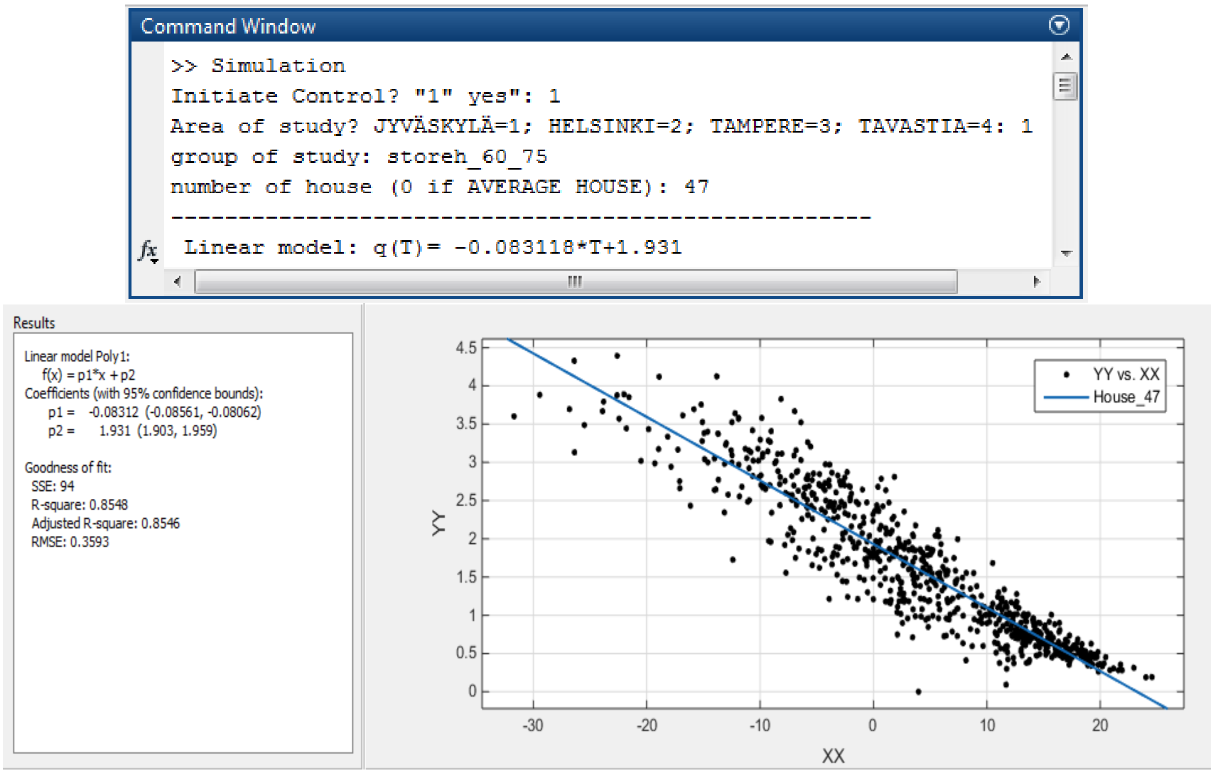


Figure 16 - Regression study

$$Hourly\ model: \qquad w_i = -0.083118 \cdot T_{i-1} + 1.931 + deviation \qquad (27)$$

This linear hourly model is used to perform the expected linear heat consumption, but as can be seen on real data plotted on top, real consumptions vary greatly from being perfect linear. Applying this model to temperature data, we generate this linear heat consumption:

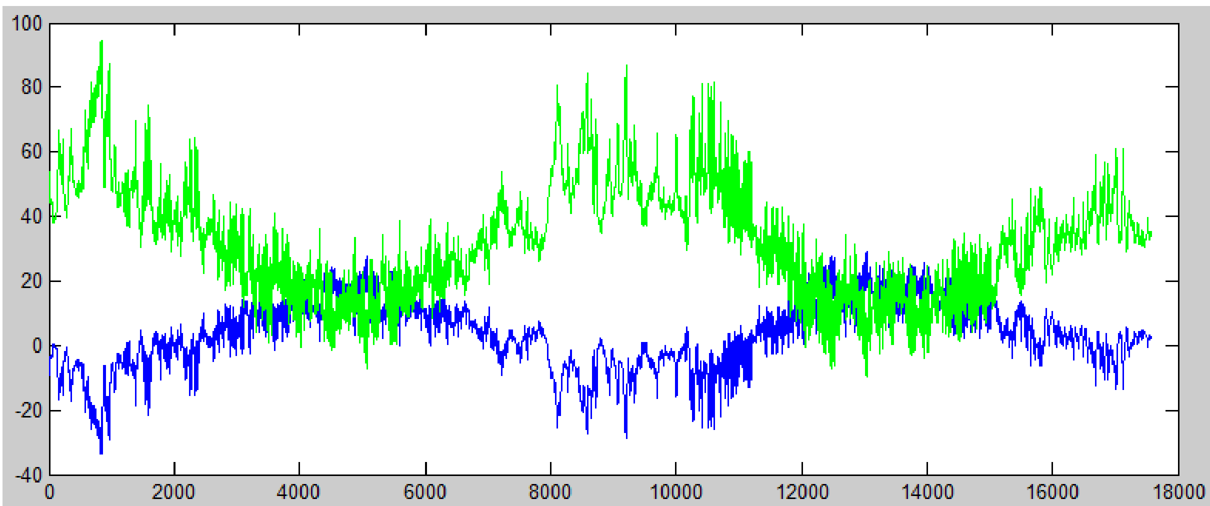


Figure 17 - This is the result after applying the linear model. In blue, the temperature's profile, scale 1:1; in green, the linear consumption's profile, scaled 20:1.

b) Deviations: errors study

As can be seen on the plot, there are deviations during ToU hours that need to be understood and forecasted somehow. They are related with the storage capacity of the house and with the stochastic behaviour of some consumptions, such as hot tap water.

For *House 47*:

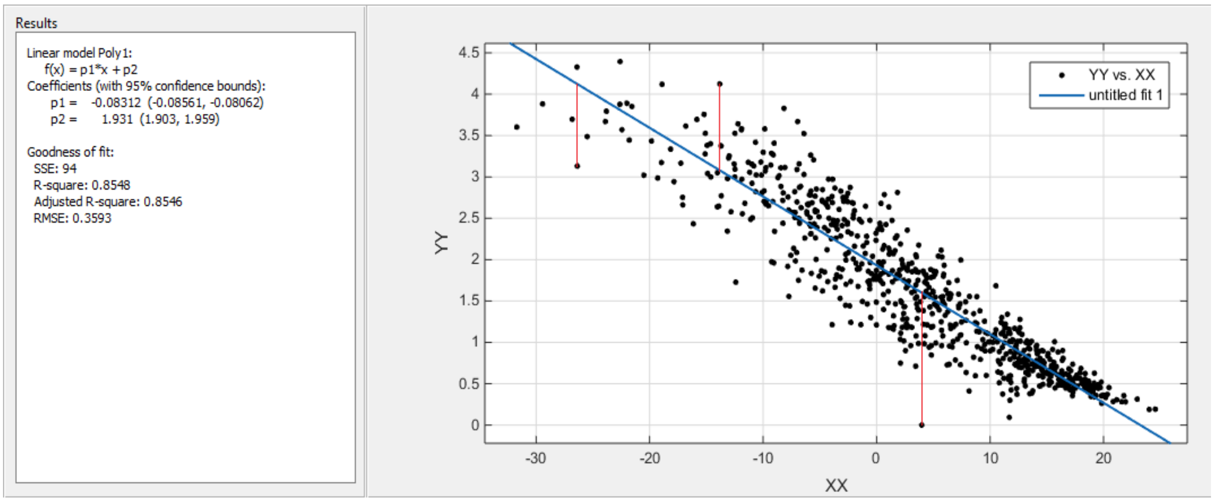


Figure 18 - Y-axis represents Hourly Consumption (kWh) whereas X-axis is average temperature outdoors on the previous day (°C).

Using the linear model, the deviations for this specific house are:

$$deviation_i = w_i - linear.model = w_i - (-0.083118 \cdot T_{i-1} + 1.931) \qquad (28)$$

$$max(\text{abs}(deviation)) = max(abs(w_i - (-0.083118 \cdot T_{i-1} + 1.931))) \qquad (29)$$

It is obtained the following maximum value of deviation in this sample house:

$$\max(\text{abs}(deviation)) = 1.2907 \approx 1.29 \frac{kWh}{h} \qquad (30)$$

This means "every hour a maximum error of $\pm 1.29 \frac{kWh}{h}$ might be committed when using the linear model to predict the consumption level". These values set the bounds between whom all real errors are distributed for this house in particular. Next step is to study the overall distribution of these real errors.

In order to fit the distribution, Normality tests are applied to the residuals. In case of being unsatisfactory, the program also applies T-Student distribution fitting and selects which fits the best for every house. Normality tests performed inside the program:

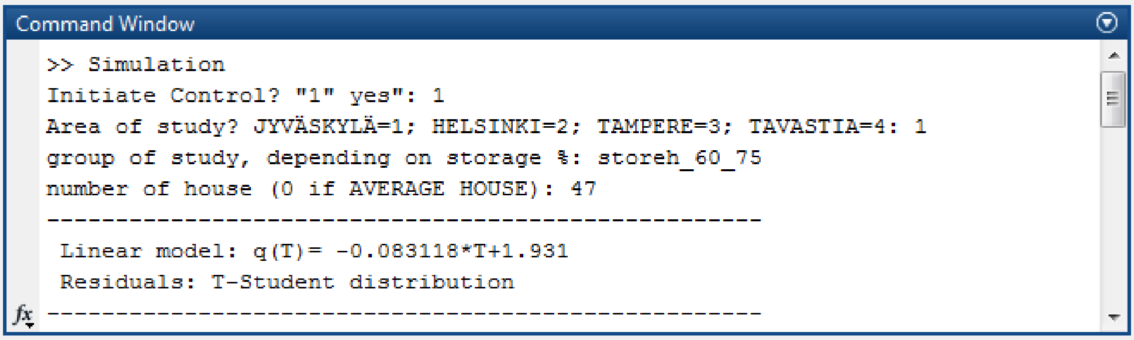
- Kolmogorov-Smirnov
- Lilliefors
- Jarque-Bera.

If one of them is passed, the program considers that the deviations are normally distributed. If none of them three is passed, T-Student analysis is run. [28]

After reproducing several tests it has been found that errors typically follow T-Student distributions in single houses, whereas when analysing average houses in different regions, Normality tests are passed successfully.

Sample: *House 47*

The distribution fails the Normality tests, so it is applied T-Student distribution fit, giving a satisfactory result (function `ttest` of Matlab®).



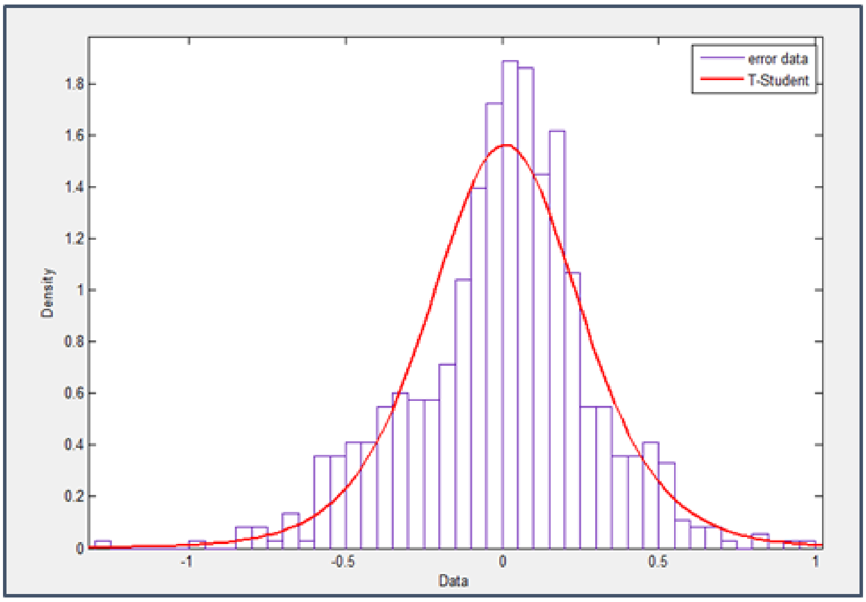


Figure 19 - T-Student distribution of historical deviations

[28] Pre-configured tests in Matlab®. They are performed in: Annex 4, *Model of Daily Consumption*.

c) Real error levels simulation

From the previous step, it has been demonstrated that errors have similar behaviour to a *T-student distribution*. In order to generate different deviation levels it is applied the *seven-step approximation* of the T-student distribution.

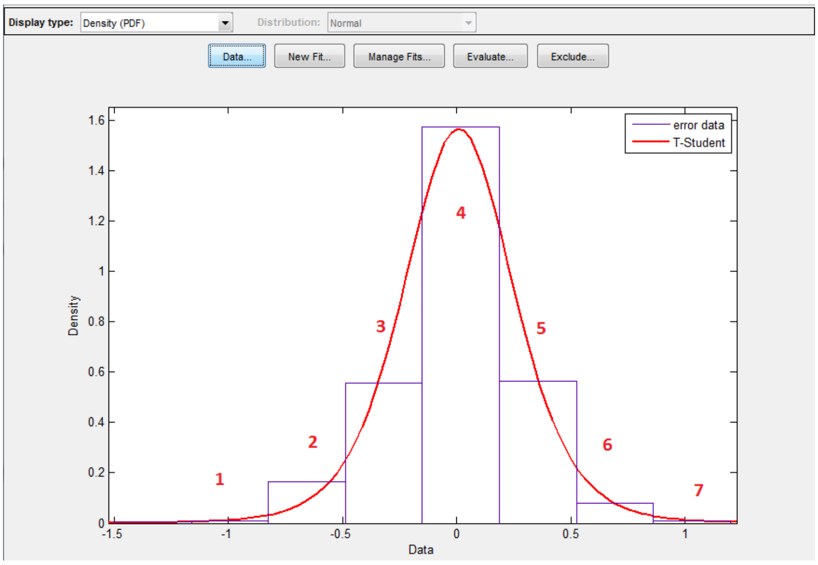


Figure 20 - 7-step approximation of the distribution

The "*seven-step approximation* of a distribution" allows to establish different error levels (7 in this case). Every level represents a deviation from the linear model with a certain cumulative probability of appearing.

The bounds of errors registered from real data in *House 47* are $\pm 1.29 \frac{kWh}{h}$, so between the range $(-1.29, +1.29)$ the program obtains 7 different levels of deviation:

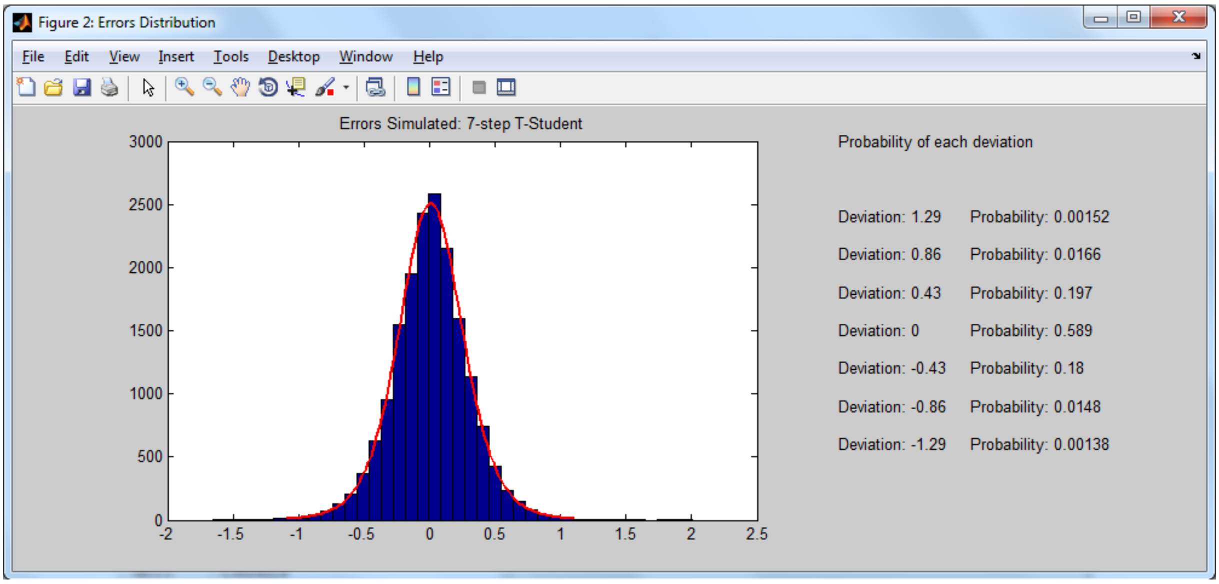


Figure 21 - On the right: first column, deviation in kWh/h; second column, its corresponding probability.

The figure shows the probability of each deviated level. The probability of having a deviation of $\pm 1.29 \frac{kWh}{h}$ is very small (0,15%), whereas a null deviation from the linear model has a probability of 58,9%. It can also be seen that symmetric levels don't present the same probability necessarily:

Table 3 - Real deviations and their corresponding cumulative probability

Real Deviation (kWh/h)	-1.29	-0.86	-0.43	0	0.43	0.86	1.29
Real Probability	0.00152	0.0166	0.197	0.589	0.18	0.0148	0.00138

Based on this distribution it can be seen that the consumption level with higher weight has null deviation from the linear model (around 59% probability to appear). Other deviated levels have lower probabilities of appearing, but are still important to take into account when creating the first population of response vectors to run the GA.

Combination of linear model and simulated errors

Next step is combining both results from (a) and (c): the linear model for the average behaviour and its possible deviations based on real errors:

$$7\ levels: \qquad \forall i \in \mathbb{N}(1,7) \quad w_i = -0.067 \cdot T_{i-1} + 1.557 + dev_i \qquad \left[\frac{kWh}{h}\right] \quad (31)$$

$$dev = [-1{,}29 \quad -0{,}86 \quad -0{,}43 \quad 0 \quad 0{,}43 \quad 0{,}86 \quad 1{,}29] \qquad \left[\frac{kWh}{h}\right] \quad (32)$$

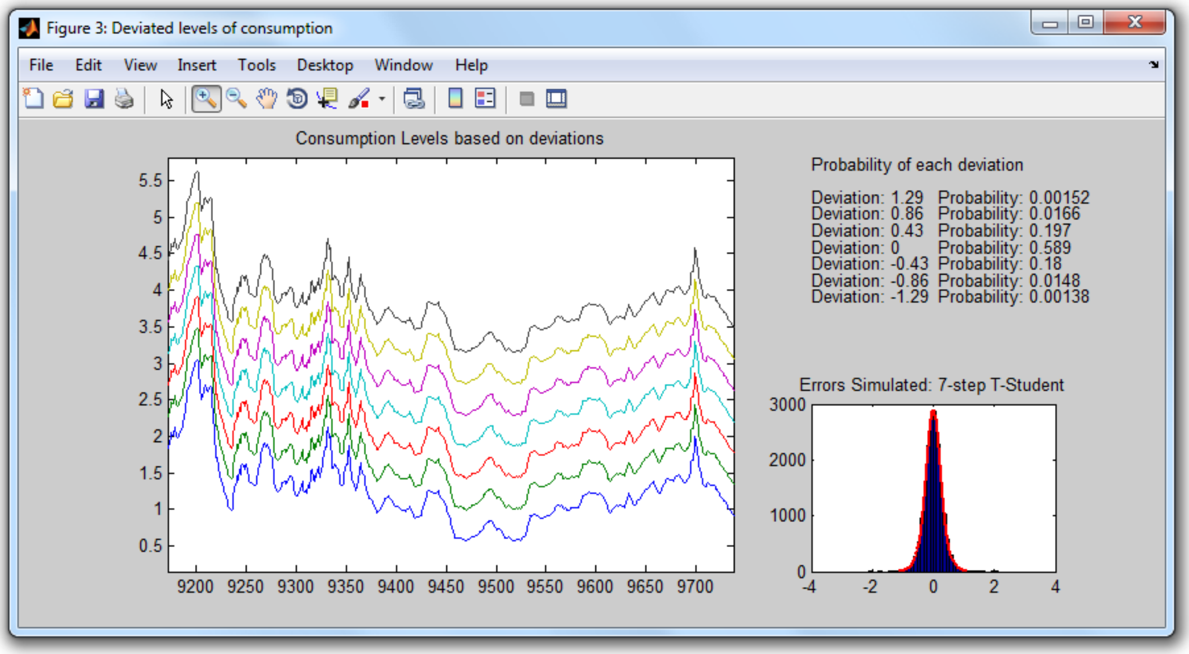


Figure 22 - (e.g. program's output: pink_cons = cyan_cons+0.43 kWh/h; green_cons = cyan_cons-0.86 kWh/h)

From the graph, cyan blue curve is the generated demand with null deviation (same values as obtained from linear model). 7 different curves, one per level of deviation.

When applying deviation to the average consumption, some values go below 0, which is an unreal situation for this kind of energy systems. These values are automatically converted to "0" by the program.

d) First Generation Response Vector (FGRV)

Once all the possible levels of consumption have been modelled, the next step is assigning a response vector of charging hours to each of them individually. The Elspot price curve of each day is known one-day-ahead: the first selection criterion is to define "loading hours" and "non-loading hours". Two criterions are used to define a cheap loading hour:

- The price of a loading hour cannot be higher than the average price of that day. This restriction is imposed for days when the average Elspot price is lower than the tariff price.
- The price of a loading hour cannot be higher than α times the price tariff. This restriction is imposed for days when the average Elspot price is higher than the tariff price.

During several days along the year, the average Elspot price of the day is notoriously higher than the tariff price. The use of α as a *price limiting parameter* restricts the selection of cheap loading hours during these days when the average price is high. In this case, it has been used a value of $\alpha = 1.5$, which leads to a maximum accepted value of $58.32\ €/MWh$ as a cheap price.

The selected tariff price is $38.88\ €/MWh$. This value corresponds to the average Elspot price between 2012 ($36.33\ €/MWh$) and 2013 ($41.16\ €/MWh$) in Finland. [14]

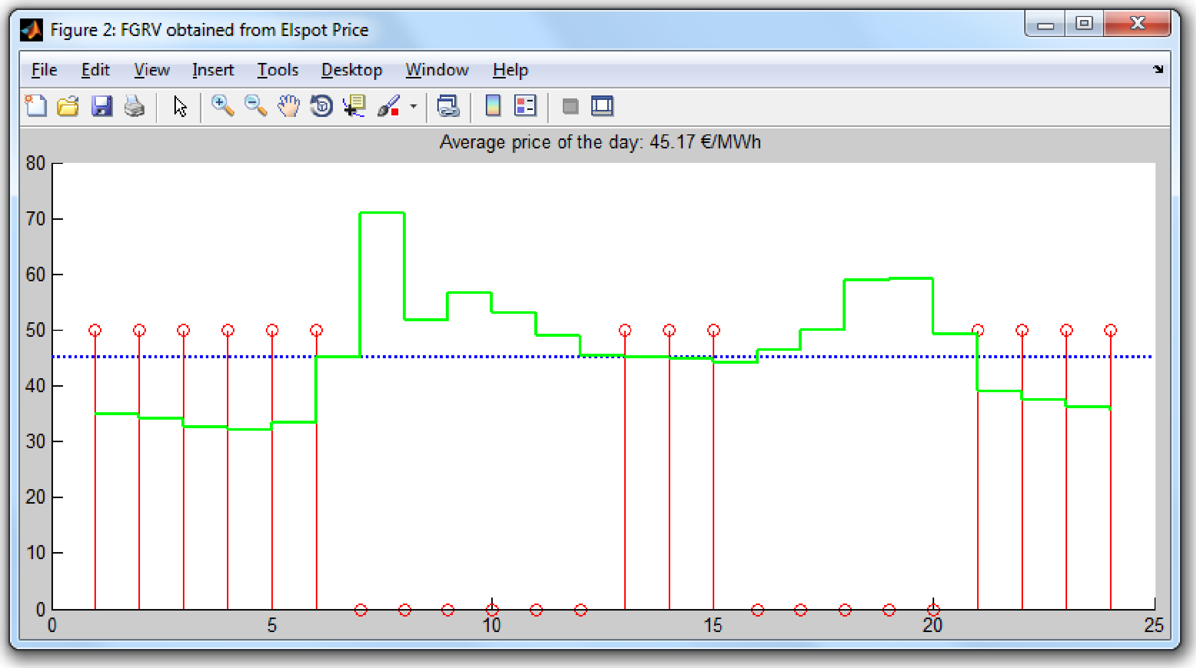


Figure 23 - First Generation Response Vector in red; Elspot curve in green; average price of the day in blue.

In red it is shown the *First Generation Response Vector* (FGRV). It only recognises cheap loading hours every single day, and it is not subjected to any physical restriction. Once the cheap loading hours are selected, the next step is to apply the formulae[29] to this vector: the result of combining FGRV and the formulae creates the next generation, the family of *Second Generation Response Vectors* (SGRV).

[29] Level of the storage tank's evolution formulae, presented in the *Mathematical Model*.

e) Second Generation Response Vectors (SGRV)

The FGRV is given as a control vector to the program, and it is automatically adjusted in order to fit the formulae[30]. In this way, the first performance of the storage level evolution (S_i) is done. For each one of the seven deviation levels, a single vector is generated by mutating its chromosomes with the following rule:

- Mutate from "1" to "0" in case of reaching 100% of capacity on the next hour.
- Mutate from "0" to "1" in case of reaching less than 20% of capacity on the next hour.

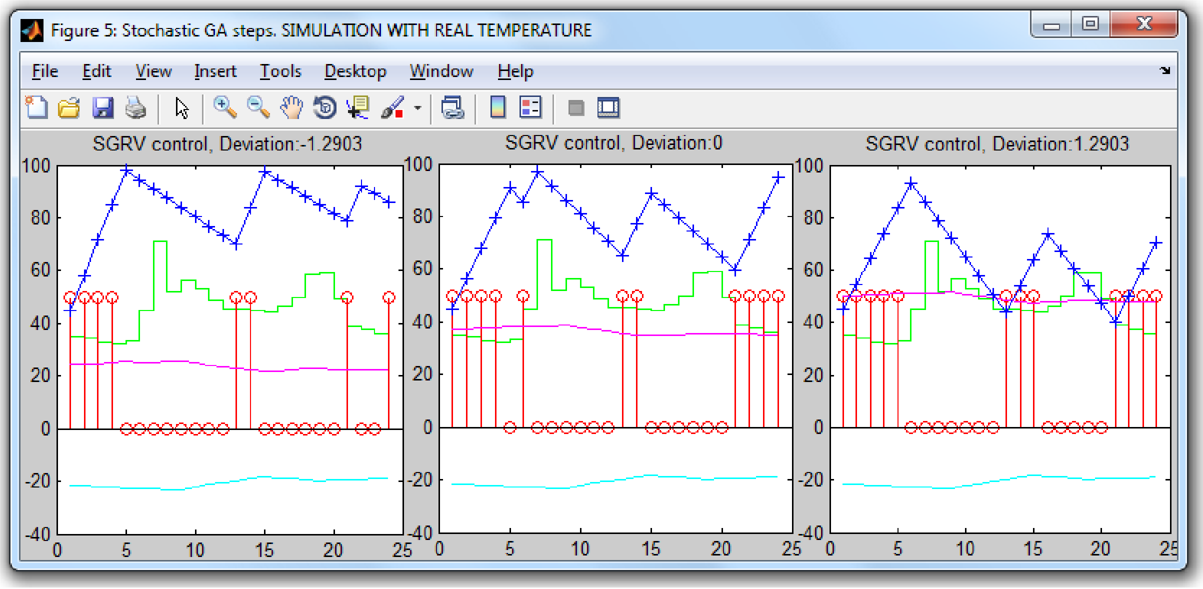


Figure 24 - 3 samples of deviated levels of consumption.

One solution is given for each of the seven deviated consumptions when the formulae are applied, forming the *Second Generation Response Vectors* (SGRV, family of seven vectors) which satisfy both physical restrictions and use of cheap hours.[31] These seven vectors look similar to the FGRV since they evolved from it, but also introduce variability in the population since they are typically different one from each other. In the picture below it can be seen a sample of three SGRV evolved from the FGRV of the previous page:

In *green*, the Elspot Price curve; in *pink*, the hourly consumption forecasted for each deviation level; in *cyan*, temperature profile of the day; in *dark blue*, tank's level simulation given by the formulae and in *red*, the adjusted response vector (SGRV).

It can be seen from the figure different SGRVs given for each consumption level (deviations of $-1.29\ \frac{kWh}{h}$, $0\ \frac{kWh}{h}$ and $1.29\ \frac{kWh}{h}$ from the average consumption). When comparing the FGRV with each of these SGRVs they can be found similarities between them, but some variations appear when the tank's level reaches the bounds: this is basically how *ToU* clients behave, loading when there is not enough energy stored independently from the price.

These solutions are not optimal yet but are closer to the optimum searched: thanks to the GA, this optimum will evolve from an initial population made with this family of SGRV.

[30] Formulae already presented in the *restrictions of the Genetic Algorithm*.
[31] Annex 7, *Internal constrains inside GA*

f) Creation of the Initial Population and GA application

The initial population of vectors is generated from this family of SGRV(e) and the probability of each deviated consumption level (c). The creation criterion is the following:

"*Generate a proportional number of individuals for each SGRV based on its probability*"

Deviation level	-1.29	-0.86	-0.43	0	0.43	0.86	1.29	**Cumulative**
Probability	0.00152	0.0166	0.197	0.589	0.18	0.0148	0.00138	1
Number of individuals	2	17	197	589	180	15	1	1001

The number inside this population is calculated as "the rounded value of 1000 times the probability of the deviation level". The value 1000 is chosen in order to create also individuals with the lowest probabilities (levels close to the consumption bounds); in this way it is obtained a representation for each possible consumption level even if it is very low. In this particular case the population is composed by 1001 individuals of SGRV.

This Initial Population is one of the inputs to the GA. The GA measures the fit of every individual of the population to the *fitness function*[32], mutates them, and then reproduce them with other individuals. The algorithm also introduces random mutations in order to study their goodness of fit: in this a way also the stochastic behaviour is introduced in the evolution process the final response vector. This may lead to reach different solutions when running the GA in different occasions with the same input, existing more than one solution accepted by the algorithm.

The reader might think that because of the number of individuals of each level of consumption, the solution given by the algorithm is always similar to the consumption with null deviation (almost 60% of the initial population). Later results will demonstrate this does not happen necessarily.

[32] Theory of Genetic Algorithm; code in Annex 6, *Fitness function inside GA*.

The following block diagram presents all the previous steps organized:

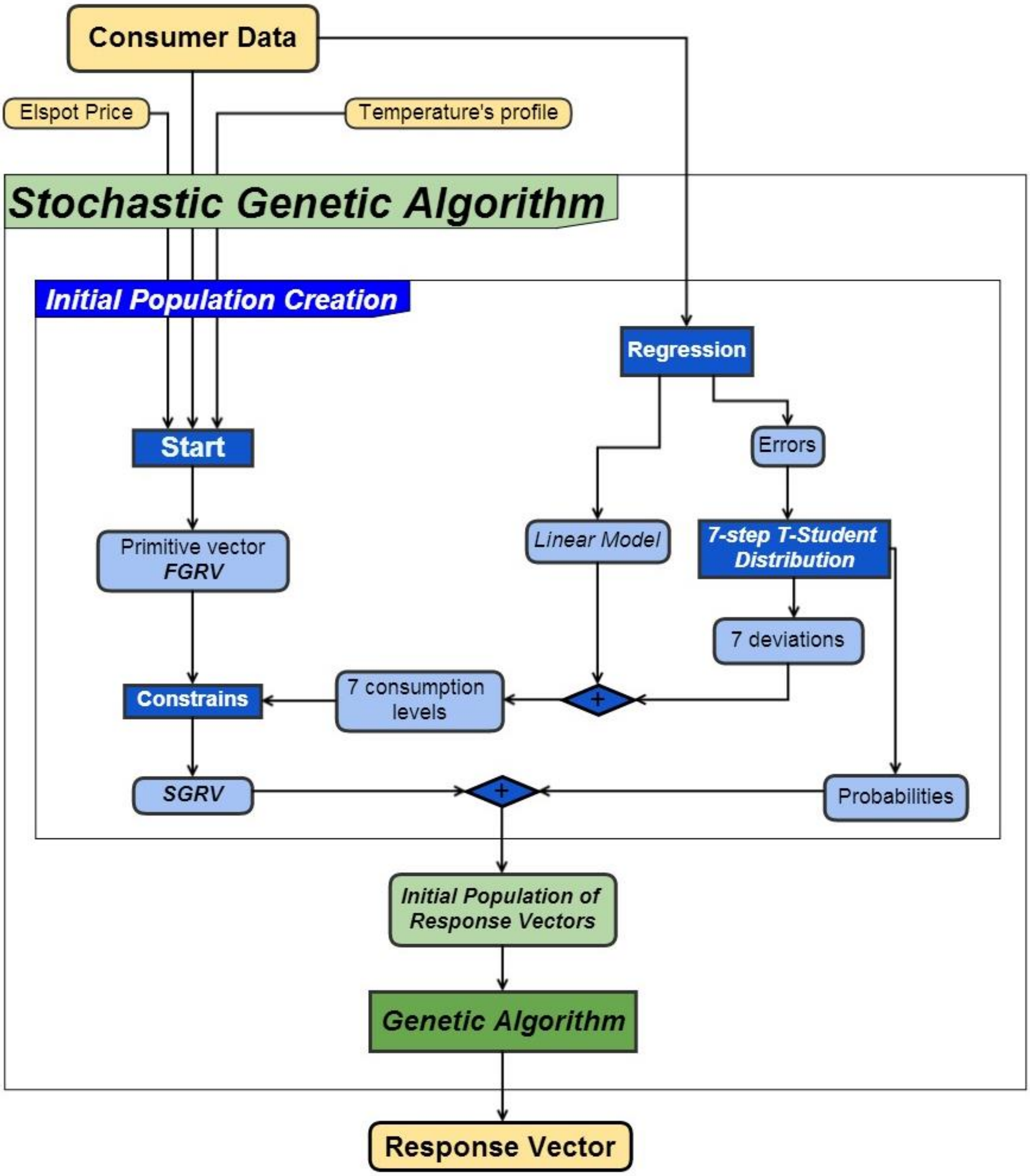


Figure 25 - Block diagram of the Stochastic Genetic Algorithm

2. Standard GA specifications & Speed Convergence Improvement

GA specifications

At GA level[33], the *Fitness function* and all *constraints* are set[34] to define the input options of the algorithm, already developed into the Optimization Toolbox[35]. This tool also allows setting parameters such as number of generations, crossover function and initial population.

GA in Matlab

```
opts=gaoptimset('PopulationSize',k,'InitialPopulation',first_gen);
[xOpt,fval,exitflag,output,population]=ga(@(x)fun(x,Elspot),24,...
        [],[],[],[],lb,ub,@(x)const_7_step(x,S,q,S_0,P,unos,dev),intCon,opts);
```

- *Population Size*: $(k = 1001)$ individuals, same as the *initial population*.
- *Initial Population*: the developed one $(first\_gen)$ [36].
- Fitness Function $(@(x)fun[\ldots])$.
- Lower Bound (lb): 24-component vector that establishes the lower bound for each chromosome. The selected lb is the the 1st SGRV, with the highest negative deviation level.

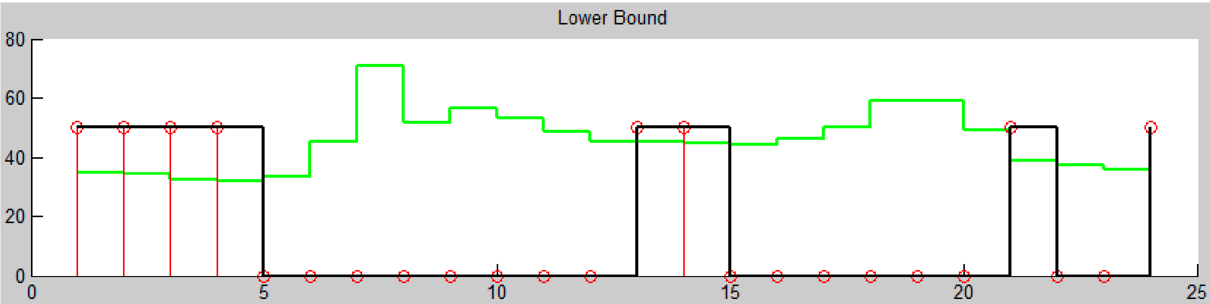


Figure 26 - In this case, lb is deviated -1.29 kWh from the average consumption.

- Upper bound (ub): 24-component vector that establishes the upper bound for each chromosome. The selected ub is the the 7th SGRV, with the highest positive deviation level.

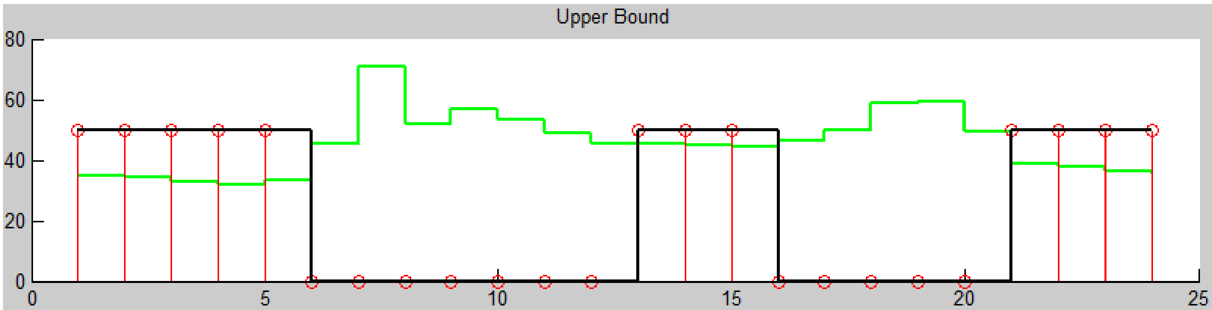


Figure 27 - In this case, 1.29 from the average consumption.

- Constraints ($@(x)const\_7\_step[\ldots]$): the physical constrains of the model.
- $intCon$: orders the GA to compute the solution with integer values (0 or 1).

[33] Consult the block diagram on the previous page.
[34] Annex 6, *Fitness function inside GA*; Annex 7, *Constraints inside GA*.
[35] Consult page 21, Genetic Algorithm's block diagram.
[36] *Creation of the Initial Population and GA application*, page 32.

Convergence speed improvement

The SGRV family is obtained from the historical analysis of the consumption record. In this family, the 1st and the 7th SGRV represent the consumption deviation bounds: the 1st is chosen as the lower bound (lb) and the 7th as the upper bound (ub).

Without the selection of these bounds, the GA looks for an optimal solution among all the possibilities. A 24-chromosome vector of binary values has $2^{24} = 16{,}777{,}216$ candidates to fit in the optimization function. By fixing lb and ub, the number of candidates is reduced notoriously. In the application case of *House 47*:

lb	**1**	**1**	**1**	**1**	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	0	**0**	**0**	**0**	**0**	**0**	**1**	0	0	**1**
ub	**1**	**1**	**1**	**1**	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	1	**0**	**0**	**0**	**0**	**0**	**1**	1	1	**1**
Fixed	X	X	X	X		X	X	X	X	X	X	X	X	X		X	X	X	X	X	X			X
Free					X										X							X	X	

Thanks to these bounds, 20 of the chromosomes are pre-selected and 4 are free, so the possible combinations have been reduced from more than 16 million candidates (2^{24}) to a group of only $2^4 = 16$ candidates.

This is how the Stochastic Genetic Algorithm improves the convergence speed of the standard Genetic Algorithm, creating a pre-selected shape of answer. There are several optimization algorithms, but the GA was chosen since it offers a wide selection of options and adjustments in order to shape the solution according to the optimization goal.

Temperature Forecast: ARMA Model

On the regression analysis it has been demonstrated that the consumption is straight related to the temperature: this means the demand of each house can be estimated by forecasting the temperature; this option is also introduced in this computational tool. Before starting an analysis, the program asks the controller if using the historical recorded temperature profile or if performing a forecast.

The forecast model used: "*The ARMA (Autoregression and Moving Average) models are used in time series analysis to describe stationary time series. These models represent time series that are generated by passing white noise through a recursive and through a nonrecursive linear filter, consecutively. In other words, the ARMA model is a combination of an autoregressive (AR) model and a moving average (MA) model.*" [36]

In this work, the ARMA model developed inside Matlab® toolbox is used to forecast the temperature of the chosen day. On the right figure they can be observed the forecast (in *blue*) and the two bounds of 95% of confidence (in *red* and *green*).

This forecast has been tested by predicting data from the record, and it has been typically found that this is a conservative model: the temperature profile forecasted is always lower than the historical one. This means the program computes a higher level of demand, giving more loading hours than the used ones afterwards: this does not interfere with the interests of the controller since they must be ready to cover all the demand independently from its size.

The aim of this work is to control the demand response and not the weather forecast, reason why any other forecasting model can be selected in order to reach more accurate results. However ARMA is a robust method that serves for this purpose.

As it has been said before, the program allows to select to perform a forecast or to use recorded temperature, giving a higher freedom when testing.

In case of selecting "forecast", since this is an iterative method of control, at the end of the day both forecast and actual temperature profile are contrasted and all the information is updated in the database.

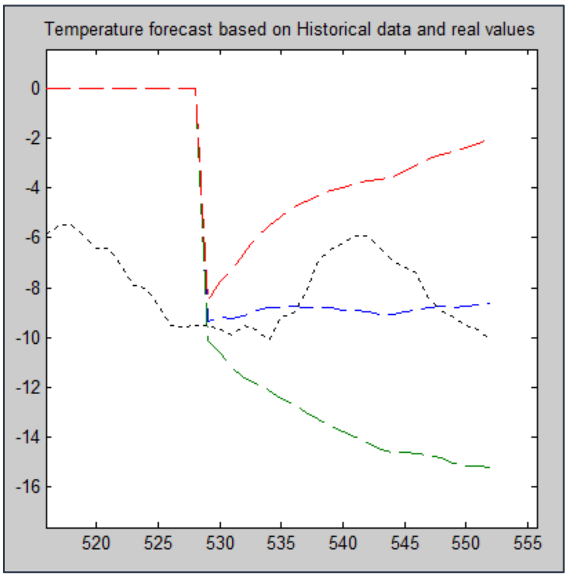


Figure 28 - ARMA model's estimation (blue) and bounds

```
Command Window
Temperature Forecast? "1" yes, "0" no: 1

    ARIMA(1,0,24) Model:
    --------------------
    Conditional Probability Distribution: Gaussian

                                  Standard          t
     Parameter       Value          Error       Statistic
    -----------   -----------   ------------   -----------
     Constant      -0.113279      0.110357       -1.02647
        AR{1}       0.983192     0.0139035        70.7155
        MA{1}         0.5813     0.0330964        17.5638
        MA{2}       0.238616       0.04301         5.5479
        MA{3}      0.0644599     0.0426342        1.51193
        MA{4}     -0.0649753     0.0513254       -1.26595
        MA{5}     -0.0802874     0.0478788       -1.67689
        MA{6}     -0.0310499     0.0448015      -0.693056
        MA{7}       0.146141     0.0540137        2.70563
        MA{8}      0.0481062      0.055172       0.871931
        MA{9}      0.0956152     0.0624688        1.53061
       MA{10}      0.0921058     0.0601317        1.53173
       MA{11}      0.0346451     0.0571229       0.606501
       MA{12}     -0.0462207     0.0590977      -0.782106
       MA{13}      0.0579688     0.0631074       0.918574
       MA{14}      0.0330793     0.0651873        0.50745
       MA{15}      0.0118272      0.059043       0.200315
       MA{16}      -0.040288     0.0571655      -0.704761
       MA{17}     -0.0120018     0.0612731      -0.195874
       MA{18}    -0.00810924     0.0671689      -0.120729
       MA{19}      -0.105199     0.0595181       -1.76751
       MA{20}     -0.0954746     0.0586423       -1.62808
       MA{21}     0.00749481     0.0578405       0.129577
       MA{22}       0.158014     0.0651771        2.42438
       MA{23}       0.195988     0.0518077          3.783
       MA{24}      0.0607379     0.0495316        1.22625
     Variance       0.180479    0.00748891        24.0996
```

Figure 29 - Screenshot of the program's question, and sample of model for House 47, day 27.

3. Output of the SGA

a) Calculations and Results

On the next page it can be found a sample of the SGA's numerical and graphical output.

1. The program performs a heating consumption plan and computes its costs of energy. The money earned is computed as the difference between the incomes from the tariff price and the costs caused by the customer to the company with this plan.
2. Studies the minimum heat requirement in *hours needed* and shows the number of loading hours given by the solution.
3. Estimates the storage level at the end of the day, and shows the number loading hours.

```
----------------------------------------------------------
          End of STOCHASTIC GENETIC ALGORITHM
----------------------------------------------------------
Money Earned with this solution: 0.15753 €
Minimum charging hours needed: 5
Number of hours offered by this solution: 8
Storage level for tomorrow: 44.8831%
Heat Consumption plan: 96 kWh
-----------------------------------------------------------
```

Figure 30 - Screenshot sample of the Stochastic Genetic Algorithm output.

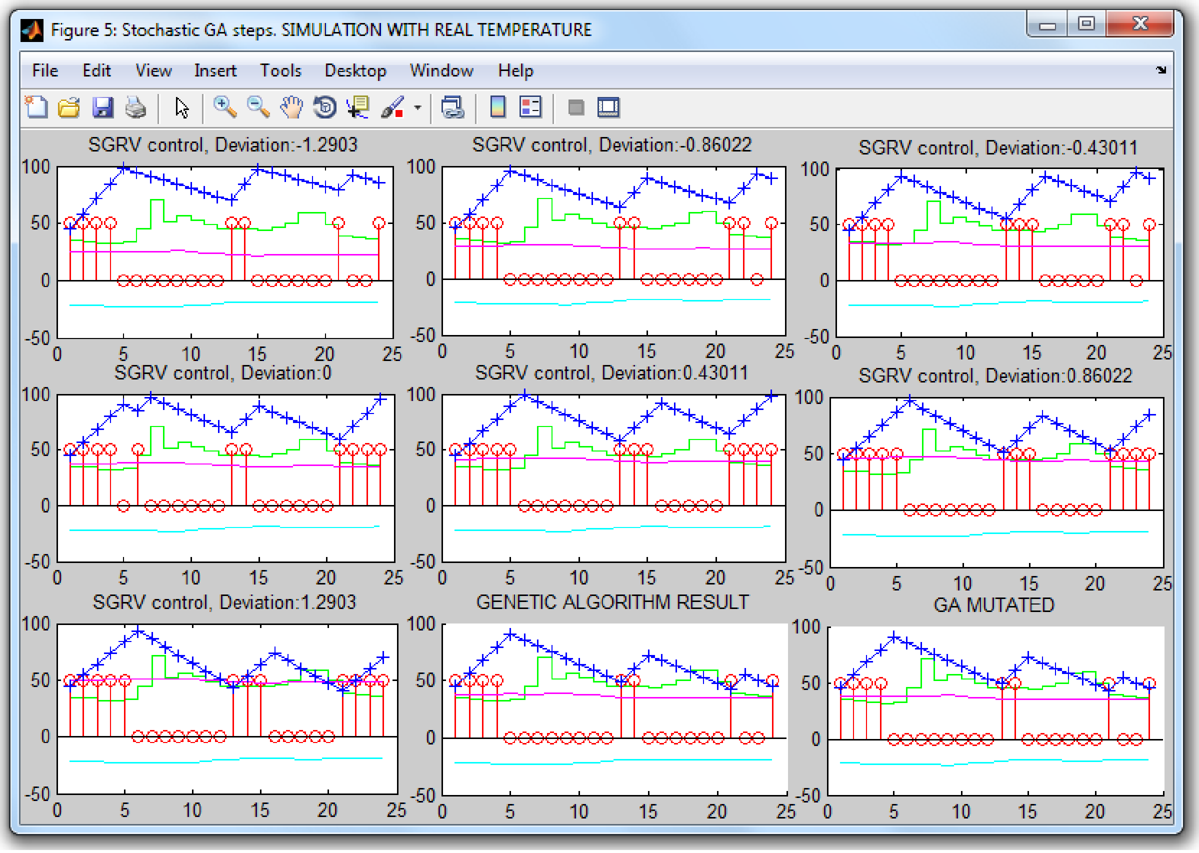


Figure 31 - Graphical output of the program for House 47. Pictures 1 to 7: the different SGRV, the 8 the solution given by the standard GA. The 9 controls that the GA solution keeps the storage between $0 \leq S_i \leq C_S$.

Shape of the response vector

The Elspot price curve follows the same behaviour from day to day. This determines a common shape for the response vector:

a. The electricity is cheap at night (traditional *ToU* hours).
b. The Elspot price shows two peak prices during daytime, one in the morning and another in the evening, caused by consumption peaks in the grid; however, it presents a valley of lower price along several consecutive hours after noon. These hours vary from day to day along the year and are fundamental for this control.

The shape of the response vector consists on a combination of the traditional *ToU* and a smart selection of daytime hours. The bigger the use of *ToU* hours, the cheaper the offer. Additionally, the selection of the cheap hours after noon. These hours vary from day to day along the year, so finding them is fundamental to this control: the profit comes by controlling these daytime loading hours.

Chapter 6. SIMULATION OF ToU BEHAVIOUR

The aim of performing a *ToU* behaviour is to simulate a level of heating consumption and a storage level evolution with it. Then a control is developed to estimate this storage level evolution, unknown to the controller, in order to perform a tracking of it.

It is important to mention that this *ToU* simulation of the level evolution is not used as data inside any calculations, keeping it unknown as its actual behaviour; it is only used for graphic purposes to show the operation of the control.

To show the validity of the program with different consumption routines, another house is used in this simulation: *House 3* from Tampere, storeh_45_60.

House details

```
Command Window
>> Simulation
Initiate Control? "1" yes": 1
Area of study? JYVÄSKYLÄ=1; HELSINKI=2; TAMPERE=3; TAVASTIA=4: 3
group of study, depending on storage %: storeh_45_60
number of house (0 if AVERAGE HOUSE): 3
----------------------------------------------------
 Linear model: q(T)= -0.14269*T+3.6042
 Residuals: T-Student distribution
----------------------------------------------------
__________________________________________________________________
                 ANALYSIS RESULTS
__________________________________________________________________
             SUMMARY OF THE HOUSE
------------------------------------------------------------------
Number of House: 3
TOTAL CONSUMPTION IN TIME SPAN: 59609.21 kWh
TANK CAPACITY C_s: 72 KWh
BOILER POWER P_B: 12 KW
TEMPERATURE: -7.0213 °C
__________________________________________________________________
```

Figure 32 - Sample of the program output

When a house is selected, the program shows on screen the linear model of the house, the result of residuals distribution analysis and the most relevant characteristics, such as C_S, P_B, *Temperature Limit* and *Total consumption in time span* (2 years in this case).

Annual simulation "downstream the storage tank"

The next step is the simulation downstream the storage tank, the real profile of heating consumption: the controller ignores what happens downstream the storage tank. The profile is created by combining the *corrected linear model*[37] and applying random deviations on the consumption.

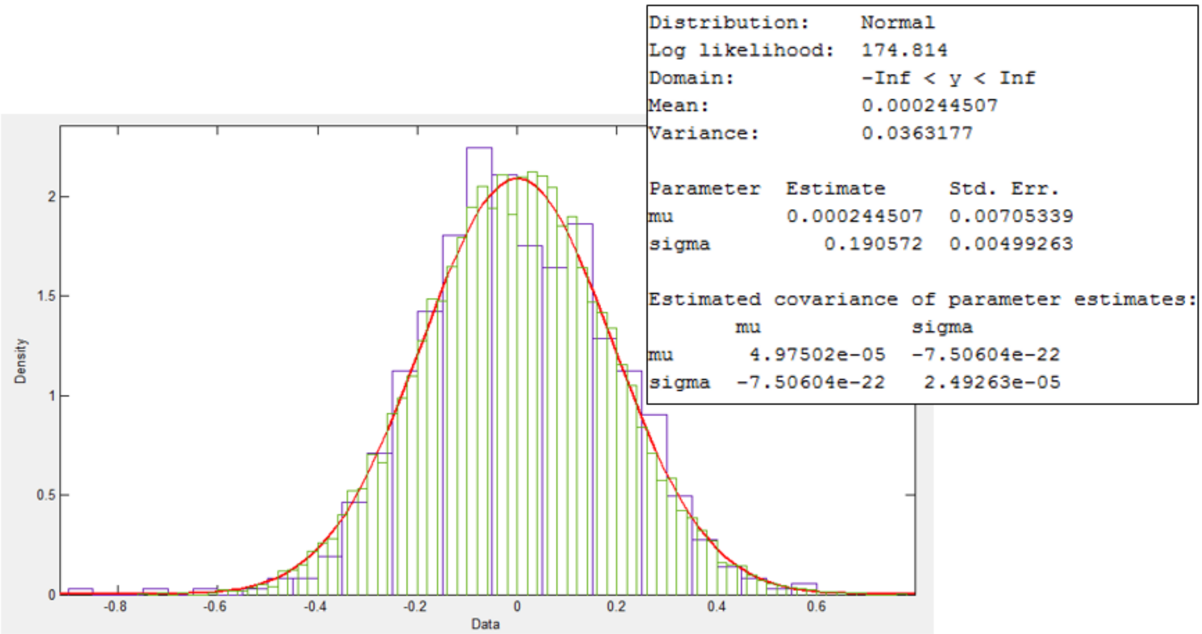


Figure 33 - In purple it can be seen the real error density distribution; in red, the Normal distribution fit of real errors; in green, random deviations generated to sum to the model.

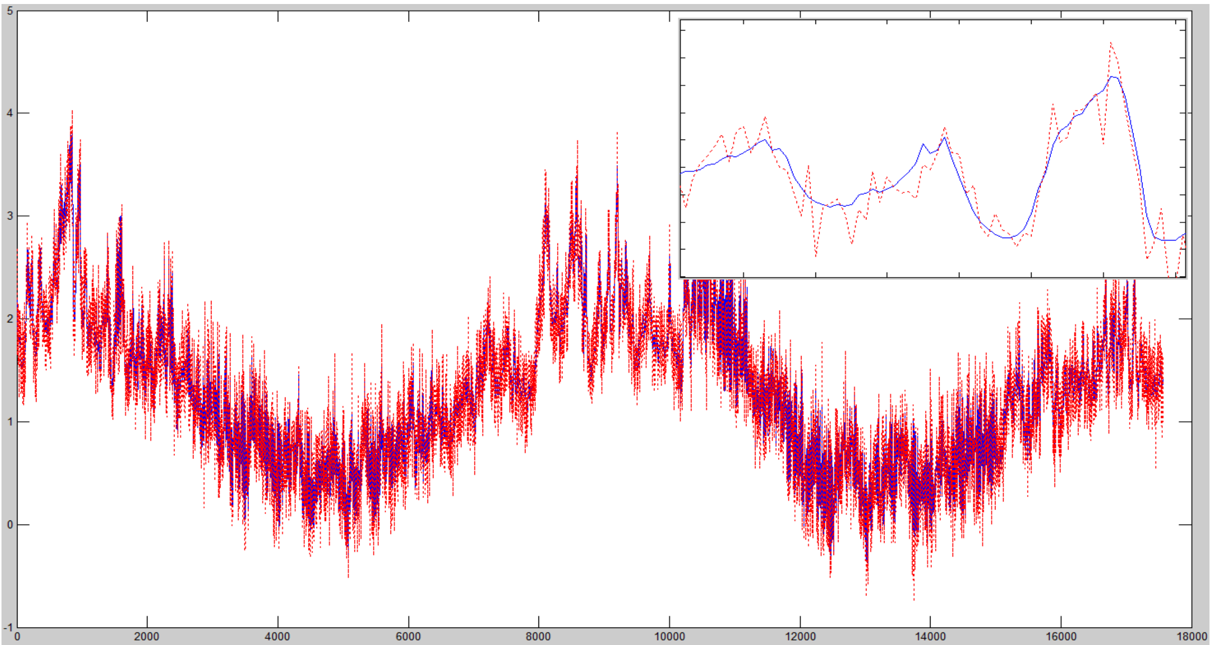


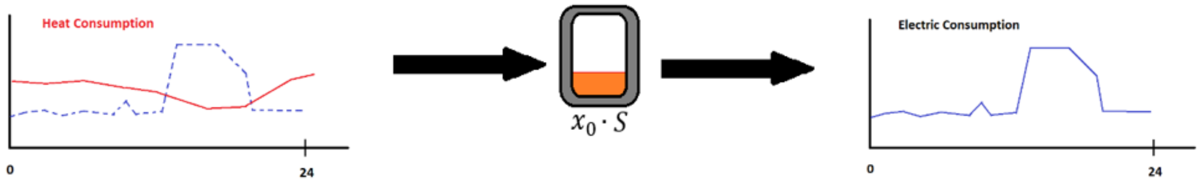
Figure 34 - The blue curve shows the heat consumption estimated from the linear model, whereas in red it is shown the simulation after applying the errors previously generated.

Thus, the consumption downstream the storage is simulated.

[37] Chapter 2, Model of daily consumption, *Level correction of the Linear Regression model*

ToU simulation “upstream the storage tank”:

Once the heating demand is simulated downstream the storage tank, the next step is to translate it into the boiler’s electric consumption. This consumption is “upstream the tank”, which means can be metered by the Distribution Company.



The procedure is the opposite of the regression model: from the simulated heat demand it is derived its corresponding electric consumption.

The first step is to create a basic ToU control vector. This is the basic pattern that the boiler follows to load:

“Give a *one* during ToU hours (from 22:00 to 7:00) and a *zero* when out of ToU”

Table 4 - Basic ToU vector.

Hour	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ToU	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Inside the program, an initial random level of storage is performed to start the simulation[38]. Then the basic *ToU* vector is applied and modified day by day in case of need.[39] The tank’s control decides whether to load during one hour or not depending on the temperature of the water inside, which means introducing variations in the ToU basic vector:

- If the storage level reaches the lower bound, the boiler switches on automatically regardless the Elspot price, which means changing a “0” by a “1” in the vector.

Table 5 - Lower bound reached at 6 pm and 7 pm, giving two extra hours of consumption.

Hour	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ToU	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	1	1

- Whereas if the upper bound is reached, the consumption is cut off.

Table 6 - Tank full at 4 am, so the boiler disconnects.

Hour	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ToU	1	1	1	1	1	**0**	**0**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

This simulation is done along the year; thus, it is obtained a ToU behaviour and a real hourly storage level evolution.

[38] Annex 11, *Total Compilation of all programs & Simulation*, *Generation of random level of storage*.
[39] Modification by following the formulae of the Mathematical Model.

Simulation results and Verification

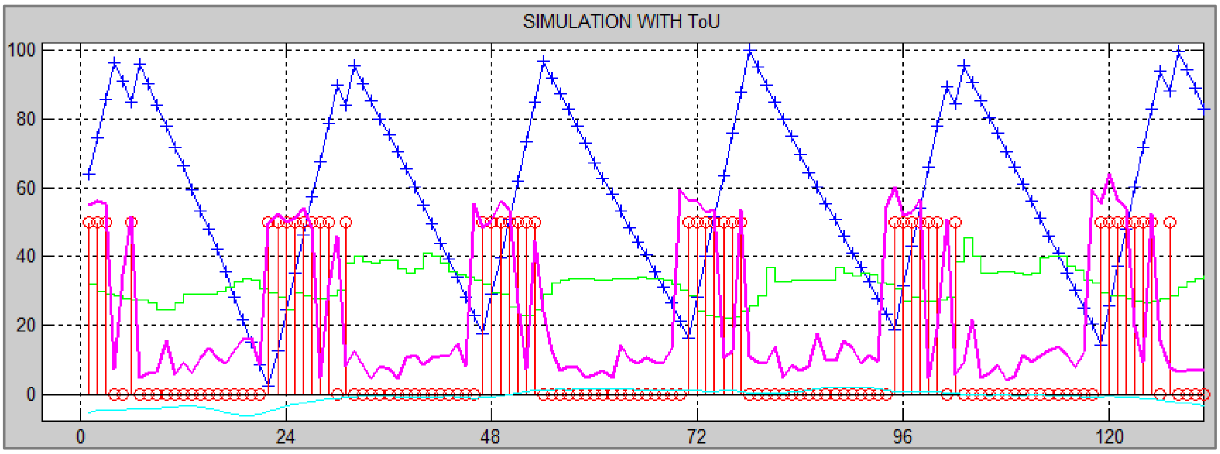


Figure 35 – Four consecutive days of ToU simulation. Contrast between real data and simulation.

On the figure above, it can be seen that typically the Elspot price is lower during night times (*green curve*), coinciding with the loading hours (in *red*) provided by *ToU*. There is no extra consumption during daytime since the storage level (in *blue*) does not get depleted. The actual consumption followed is shown in *magenta*. The *Temperature Limit* of this house is −7.0213°C[40]; in the previously plotted days the temperature does not go below this value. This simulation behaves according to the expectations. The next situation studied corresponds to several days in which the temperature goes below the *Temperature Limit*:

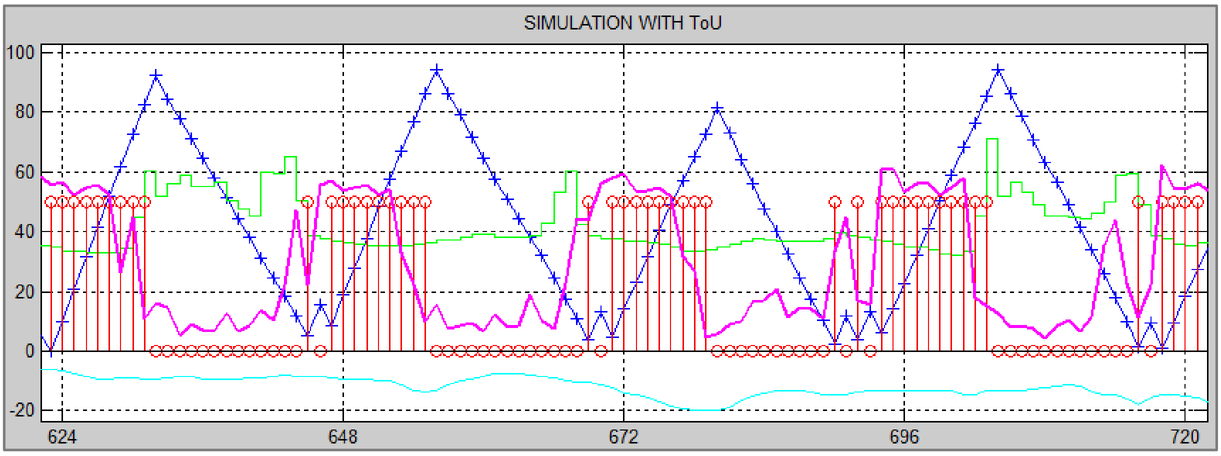


Figure 36 - Four consecutive days of ToU simulation. Contrast between real data and simulation..

Here it can be appreciated that the consumption is higher out of *ToU* hours. The simulated *ToU* (in *red*) and the actual consumption (in *magenta*) are similar. On the next page both consumptions are numericaly contrasted.

[40] See the table on page 39.

Verification of the simulated consumption level

By doing visual analysis, it can be thought *a priori* that the *ToU* simulation gives a higher level of consumption than the metered one: this motivates the measurement and comparison of both. The simulation has the following consumption level:

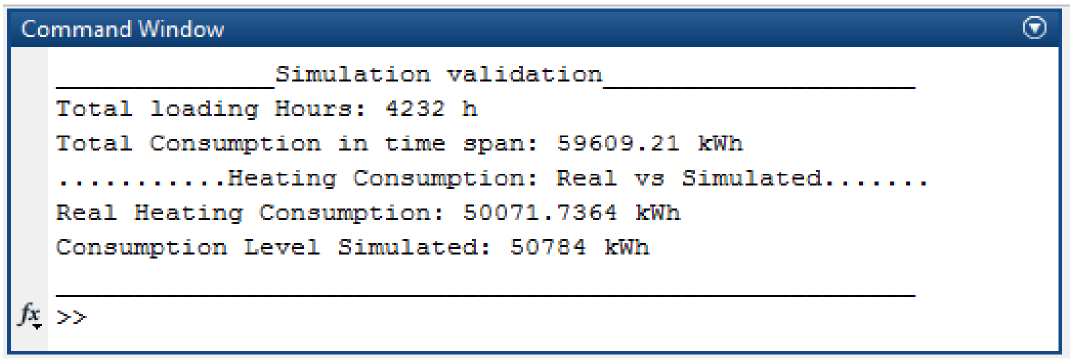


Figure 37 - Verification of the simulated consumption level

As can be seen, the solution gives 4232 loading hours, a 24,6 % of record's time span (total amount of hours in two years, 17568 hours).

During all these hours it is assumed that the boiler loads at its maximum power. The boiler power computed for this house:

$$P_B = 12\ kW \tag{33}$$

Combining both results, the estimated annual heating consumption of this simulation is

$$4232\ h \cdot 12\ kW = 50784\ kWh \approx \mathbf{50,8\ MWh} \tag{34}$$

From metered data it is integrated the whole electric consumption in time span. It is recorded inside the program as *sum of Consumption*:

$$Sum\ Consumption = 59609\ kWh \tag{35}$$

And assuming that the 84% of it is consumed for heating [4]:

$$Heating = \text{Sum Consumption} \cdot 0.84 = 50072\ kWh \approx \mathbf{50\ MWh} \tag{36}$$

It is demonstrated the validity of the simulation since both values are similar.

Chapter 7. CONTROL TOOL

This tool allows the controller to estimate the storage level at the end of the day.[41]

Before the day of the control starts, the controller performs a weather forecast in order to derive the heating consumption of tomorrow; the Elspot Price is also known beforehand, so by using the *SGA* it can be performed the cheapest loading hours' pattern. But at the same time, the storage level (one of the inputs of the *SGA*) is still unknown.

From this forecasted heating demand it is created a *one-day-ahead consumption plan*. According to the accuracy of the forecast, the customer's actual demand fits this vector pattern or introduces variations on it; this demand is metered.

The controller ignores the storage level, but knows how the boiler switches: if the energy stored gets depleted, the boiler switches *on* automatically regardless the given loading pattern, whereas if the storage level reaches the maximum storing capacity, the boiler switches *off* also independently from the control given.

A house is chosen to start its control. The controller selects manually an initial storage level[42], virtual, and performs a simulation of the storage tank evolution one day in advance. With the simulation, the controller obtains a control vector. This control vector is provided to the customer, who uses it according to its needs. Once the day is over, the controller possesses two different vectors as consumption profiles: the *consumption plan* given one day in advance (vector pattern) and the *real metered demand*, showing how the customer fit the control.

The *control tool* works by contrasting both consumption profiles, looking for the variations introduced by the customer to the control given and interpreting why they happened. Two variations can appear between both vectors:

a) The *one-day-ahead* vector offers a 1 at a certain hour ("load now") and the customer does not use it, which means the boiler remains switched *off*. This can only occur because the tank is already full and cannot charge anymore. At the end of the day the controller can recognise that, at this hour of the day, the storage was close to its 100% of capacity.

b) The *one-day-ahead* vector offers a 0 at a certain hour ("do not load") and on the record it appears a high consumption, which means the boiler is switched *on* and loading. This occurs because there is not enough energy stored to cover the demand and the system needs to charge. At the end of the day the controller can recognise that, at this hour of the day, the storage was close to be depleted.

All these variations are recorded at the end of the day, but only one is interesting to the controller: the last variation, since it is the closest one in time to the actual hour. At this point also the actual temperature profile is known, so the forecasted temperature can be rejected in order to use the actual temperature profile on the following calculations.

[41] Annex 12, *Results Validation, Control of capacity*.

[42] The starting storage percentage selected is 20%, the lower bound; explained on the next page.

Then, the storage level control is run to update the storage evolution simulation used by the controller:

1) "If the last registered variation is "*a) type*", update the storage level of this hour with the following value":

$$S_{change\ 1} = Capacity - Boiler\ Power = C_s - P_B \quad (37)$$

2) "If the last registered variation is "*b) type*", update the storage level of this hour with the lower bound": [43]

$$S_{change\ 2} = 0 \quad (38)$$

By selecting those values of $S_{change\ 1}$ and $S_{change\ 2}$ it is ensured that the real storage level remains above this reference value:

1) By applying energy balance, the maximum power input per hour to the heating system is the boiler power (P_B). When the boiler switches *off* in a "loading hour" it means the storage level has reached the upper bound.

   If on the previous hour the storage level was lower than $S = C_s - P_B$ (100% of capacity minus power input), the boiler would load also this hour with an energy input of P_B and the control tool would not detect any modification from the given pattern; but since the consumption pattern is changed in order not to load it can be ensured that on the next hour it is reached 100% with the *one-day-ahead* plan. Once the variation is introduced on the vector, it can be ensured that the actual storage level is higher than $S = C_s - P_B$ during that hour. It is a conservative solution that leaves the controller's level under the actual value.

2) On the other hand, when the boiler switches *on* in a "not loading hour" it means the storage is depleted during that hour, or that the energy stored was not enough to cover the demand.

Once the variation is detected, this value is updated and the storage level evolution is run with the actual temperature profile from this hour; if there is no variation metered, the simulated evolution of storage level is considered as valid. Thus, a better approximation of the real storage evolution is reached and also it is got a reference value of storage level to start the control of the following day.

The control is described on the following pages *step-by-step* to understand visually the working procedure of this tool. Several examples are run to present the different possibilities that the program can handle in terms of varying the vector pattern. It is used the *ToU* simulation previously introduced to create a storage level evolution to track.

Then, a *total control* is performed to *House 5*, Helsinki and *House 9*, Tavastia.

[43] The selection of both values will be presented and defended on the following pages.

Step-by-step Control

a) Starting data:

- Consumption record of the house
- Storage capacity (C_s)
- Linear regression model of hourly heat consumption
- Elspot price curve known one-day-ahead
- Temperature forecast 24 hours in advance.

b) Control starts: *One-day-ahead* forecast. Different variations detection

i. No variation between the control vector and the demand.

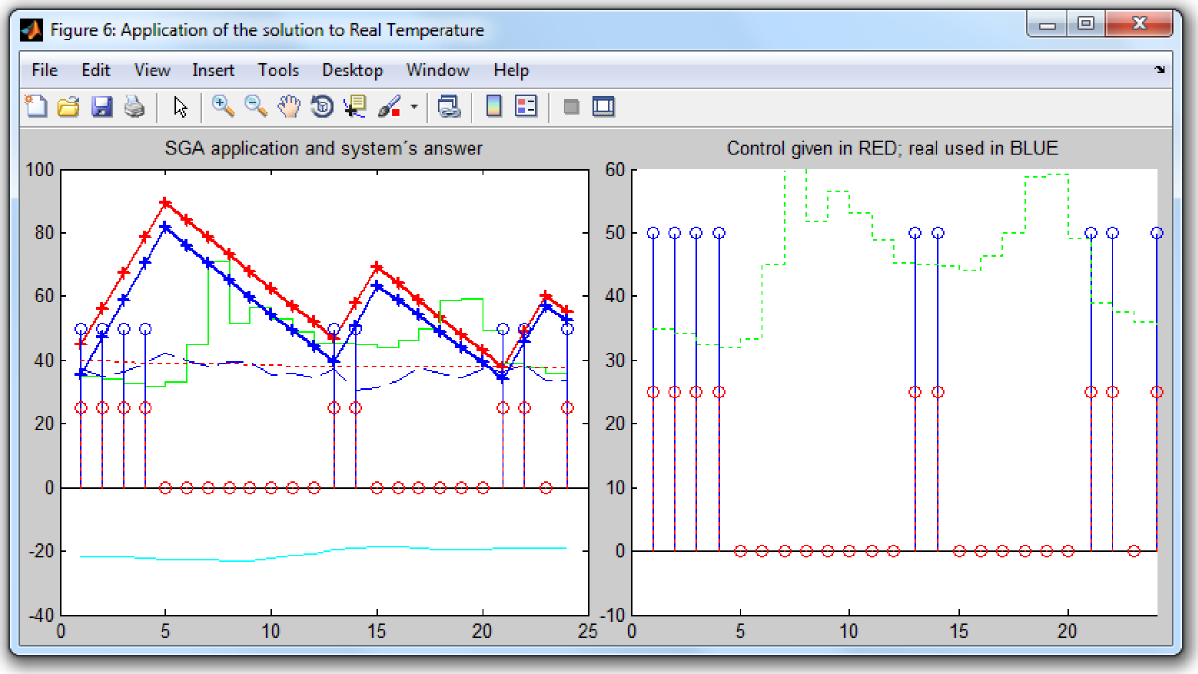


Figure 38 – One-day-ahead Control. Case 1: no variation metered.

On the left picture, in *red* it can be seen the estimation made by the controller one day in advance with the temperature forecasted; in blue, the actual behaviour of the customer during the selected day. This blue behaviour is never seen by the controller, but here it is plotted to understand how the control tool fits the storage level. It can be noticed that both temperatures profiles vary a bit one from the other since the slope of both evolutions is different.

On the right picture, the control vector in *red* and the real vector used in *blue*. The controller cannot see the actual level evolution, but he can record the consumption and derive the *blue* vector from it. Once he owns both vectors, he can detect variations from the control. In this particular case, there is no variation registered by the controller.

Thereby, when the customer follows the control vector without introducing any variation, the day-ahead estimation of the storage evolution is considered as valid.

ii. Lower demand than the consumption plan: upper bound.

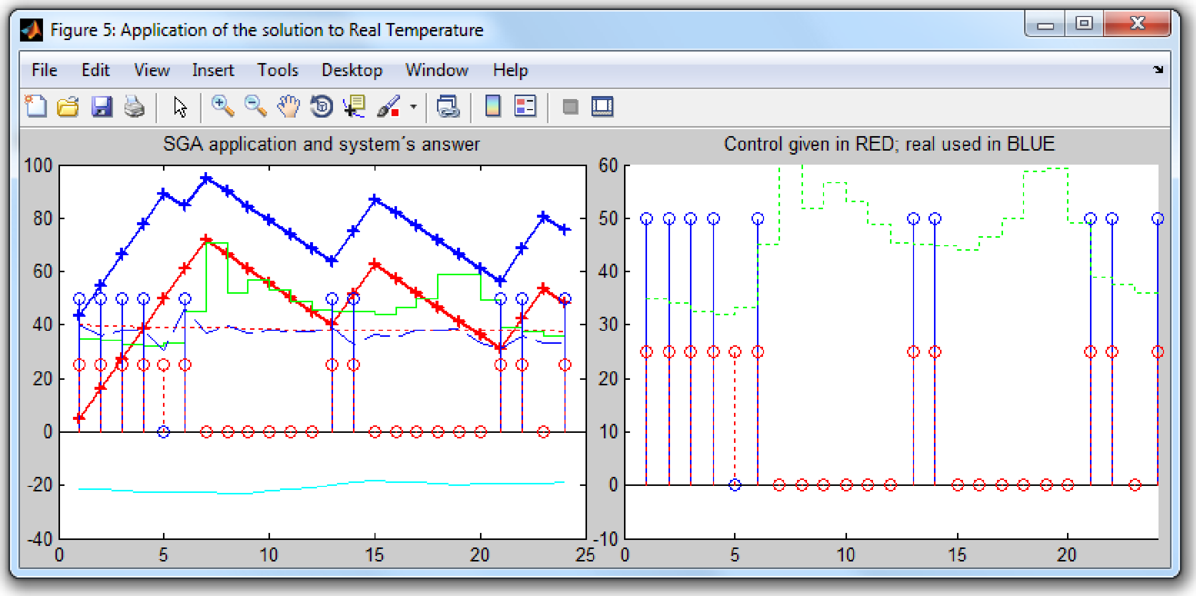


Figure 39 - One-day-ahead Control. Case 2: variation b)

In this case, the last variation occurs at 4 am, when the control (*red*) gives a loading hour that the house does not use. This is registered as "the storage is full at 4 am".

iii. Higher demand than the consumption plan: lower bound.

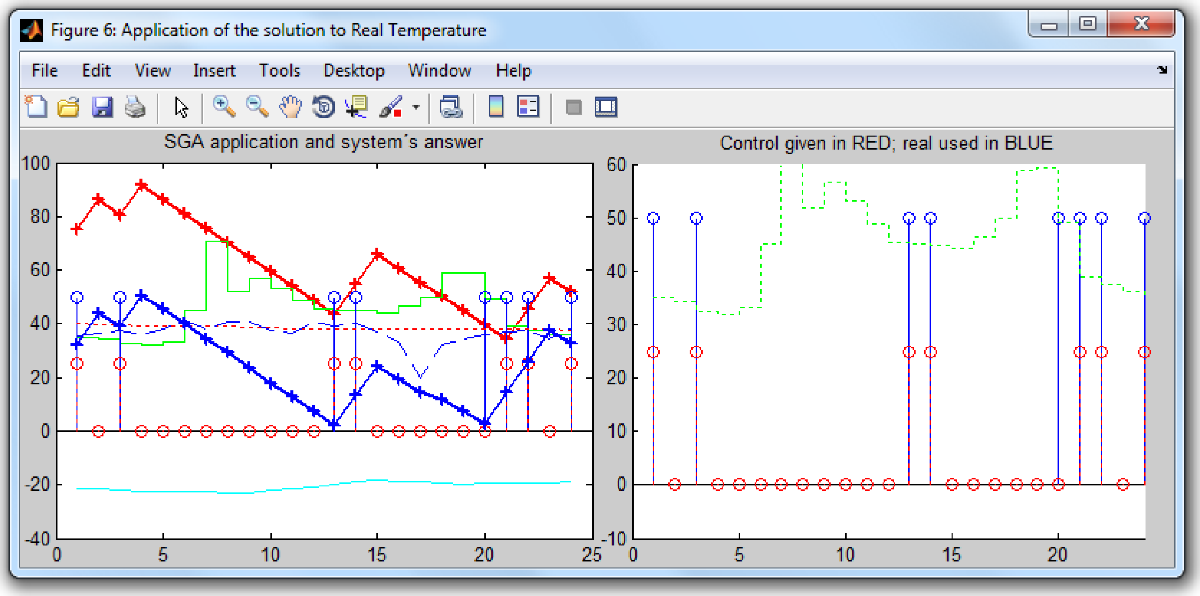


Figure 40 - One-day-ahead Control. Case 2: variation b).

In this situation, at 8 pm the control (*red*) does not offer a loading hour (0) but the boiler starts loading. This is registered as "the storage is depleted at 8 pm". It can be appreciated on the left figure (blue curve), but this curve is unknown to the controller.

NOTE: These three different possibilities that appear on the graphs (in *red*) still keep the forecasted temperature profile because the day is not over yet: that is why the slope of both exhibit different evolutions.

c) The day is over: storage level update.

i. No variation between the control plan and the demand.

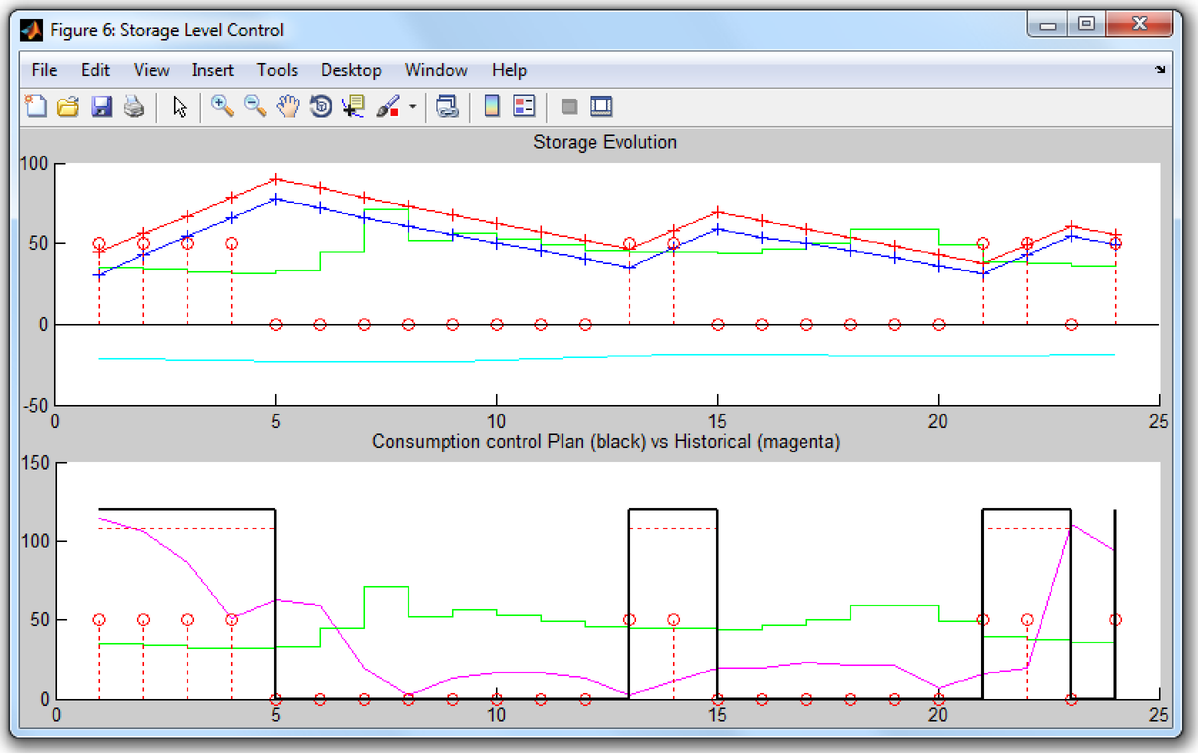


Figure 41 - There is no update of the storage level since there is no variation registered.

ii. Higher demand than the consumption plan: storage depleted.

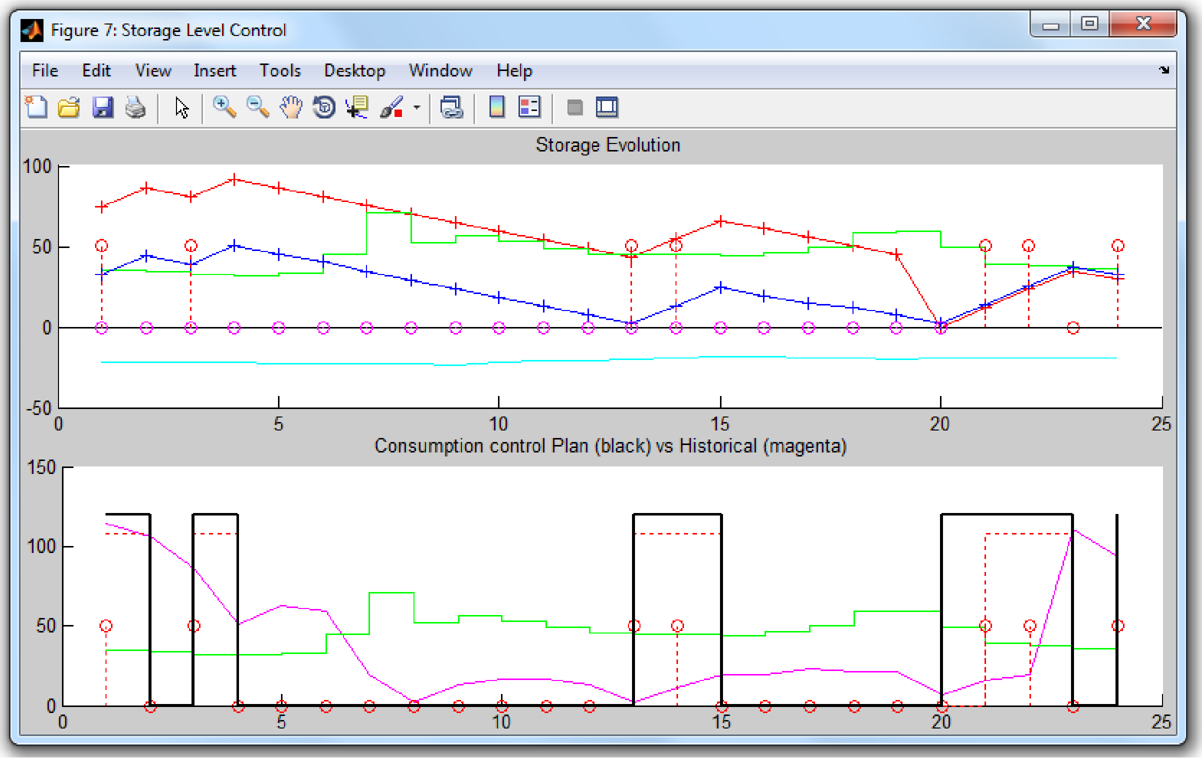


Figure 42 - Control evolution (red) updated at 8 pm: storage depleted. Detection in purple.

iii. Lower demand than the consumption plan: upper bound.

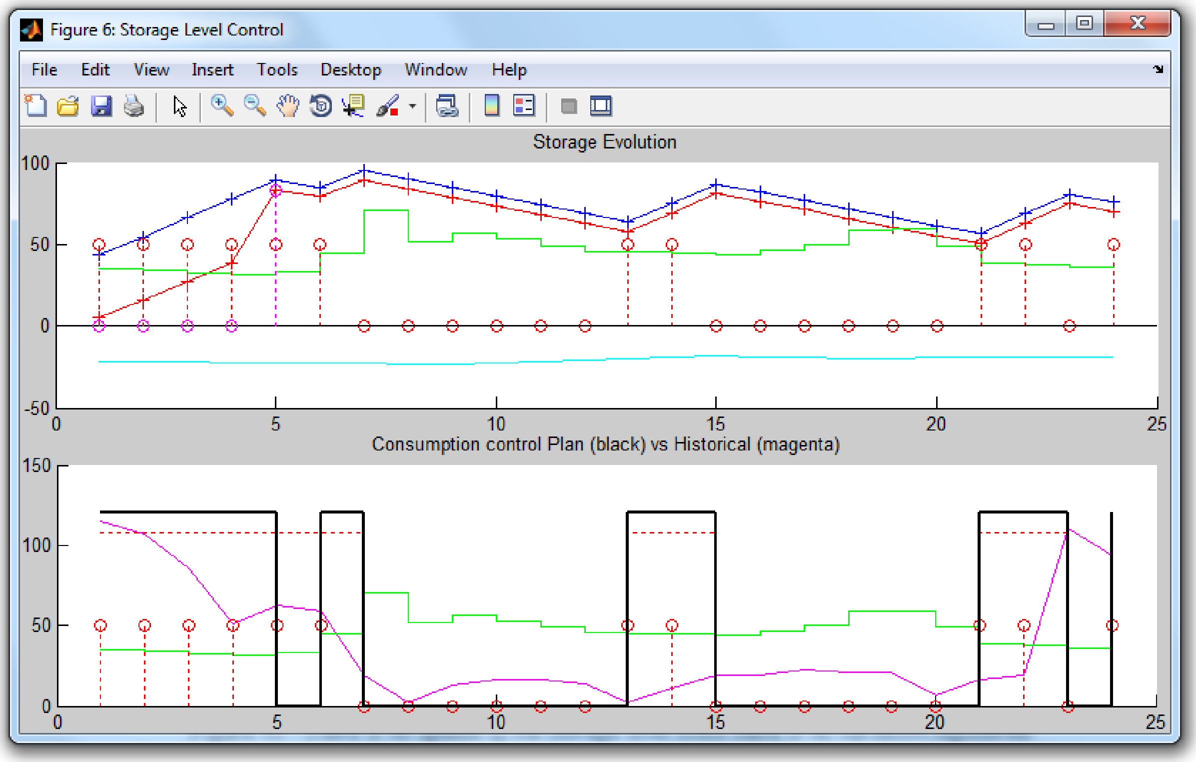


Figure 43 - Control evolution (red) updated at 5 pm: storage full. Detection in purple.

On the first case there is no control update of the storage level. The estimation is considered good: the program records the last estimation of S_{24} as the starting point of the storage level for the next day.

On the other hand, on the second and third graph there is a variation registered: the variations detector is shown in *magenta* on the upper graph of the figures. The control updates this value according to the variation type, and starts computing the storage level evolution of the following hours with the actual temperature profile. This can be seen on the picture above where from 4 am, when the variation was detected, both the real level evolution and estimation follow the same trend.

The control does not give the exact level of storage, but ensures it is bounded below by the estimation when it is updated. By using a lower value than the actual one on the calculations the controller ensures to satisfy the customer's demand for the following day: the solution given by the SGA offers more loading hours to the customer, who will selects them according to his needs.

Chapter 8. CONTROL PROGRAM

The *control program* is the final compilation of all the tools and models developed in this work[44]. In this chapter, all the different steps of the control are presented and interpreted.

It is performed to two different houses in order to study its validity: *House 5*, *storeh_45_60*, Helsinki and *House 9*, *storeh_60_75*, Tavastia.

The control is run on a selected day, and then the house is tracked also during the following day to check the program's outcome once the storage level is under control.

House 5, Helsinki

House details, models and graphical outputs

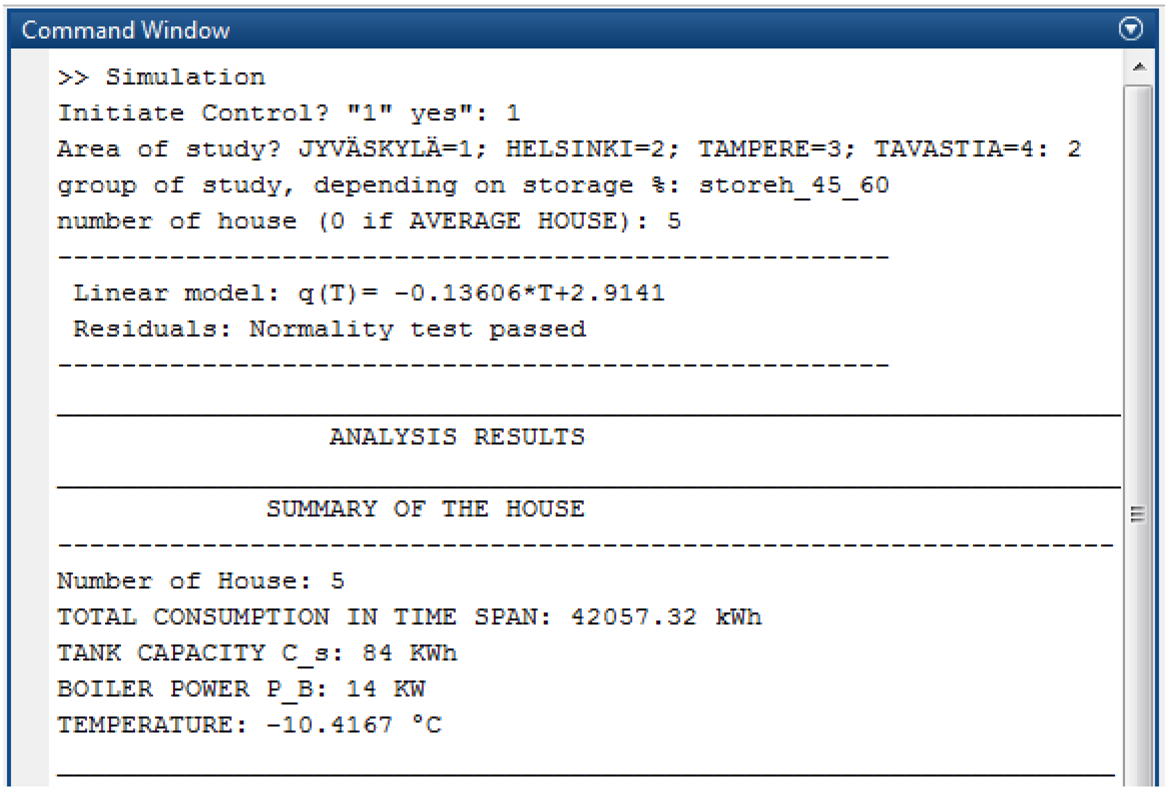

```
Command Window
>> Simulation
Initiate Control? "1" yes": 1
Area of study? JYVÄSKYLÄ=1; HELSINKI=2; TAMPERE=3; TAVASTIA=4: 2
group of study, depending on storage %: storeh_45_60
number of house (0 if AVERAGE HOUSE): 5
-----------------------------------------------------
 Linear model: q(T)= -0.13606*T+2.9141
 Residuals: Normality test passed
-----------------------------------------------------

                 ANALYSIS RESULTS

             SUMMARY OF THE HOUSE
-------------------------------------------------------------------
Number of House: 5
TOTAL CONSUMPTION IN TIME SPAN: 42057.32 kWh
TANK CAPACITY C_s: 84 KWh
BOILER POWER P_B: 14 KW
TEMPERATURE: -10.4167 °C
```

Figure 44 - Control simulation

First, the program shows on screen the analysis results and summary of the house details. This house has passed the normality test on deviations distribution, and its linear model is presented. In *Summary of the house* there are shown the most important references of the house.

[44] Annex 11, *Total Compilation of all programs & Simulation*.

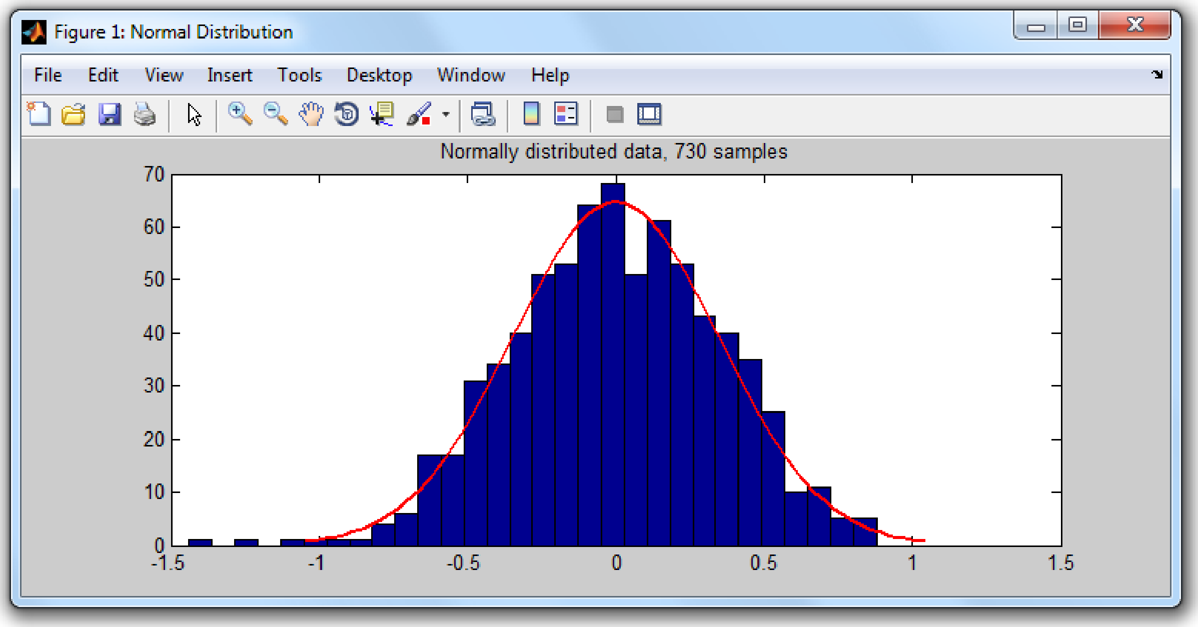


Figure 1: historical error distribution of the deviations from linear model (in the *blue* diagram) with the normal fit (in *red*).

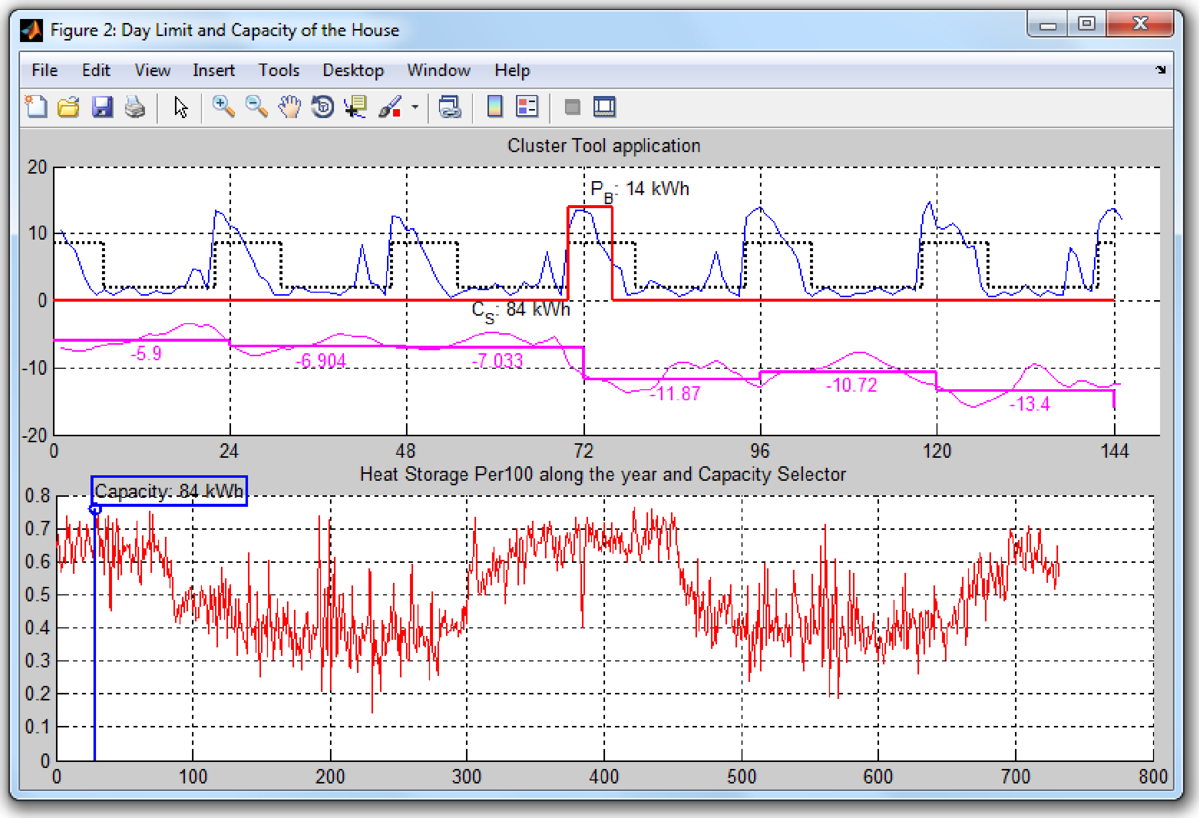


Figure 2: Control of capacity from metered data. On the first subplot it is shown the performed pattern (in *black*) and the selected day with the estimated storage capacity ($C_s = 84\ kWh$) above it. It is interesting to see that the program selected this specific day because it fits the pattern, whereas the previous and following day, with lower temperatures, exhibit a peak of consumption in the afternoon. The second subplot is a historical record of heat storage percentages, highlighting the position and value of the selected one.

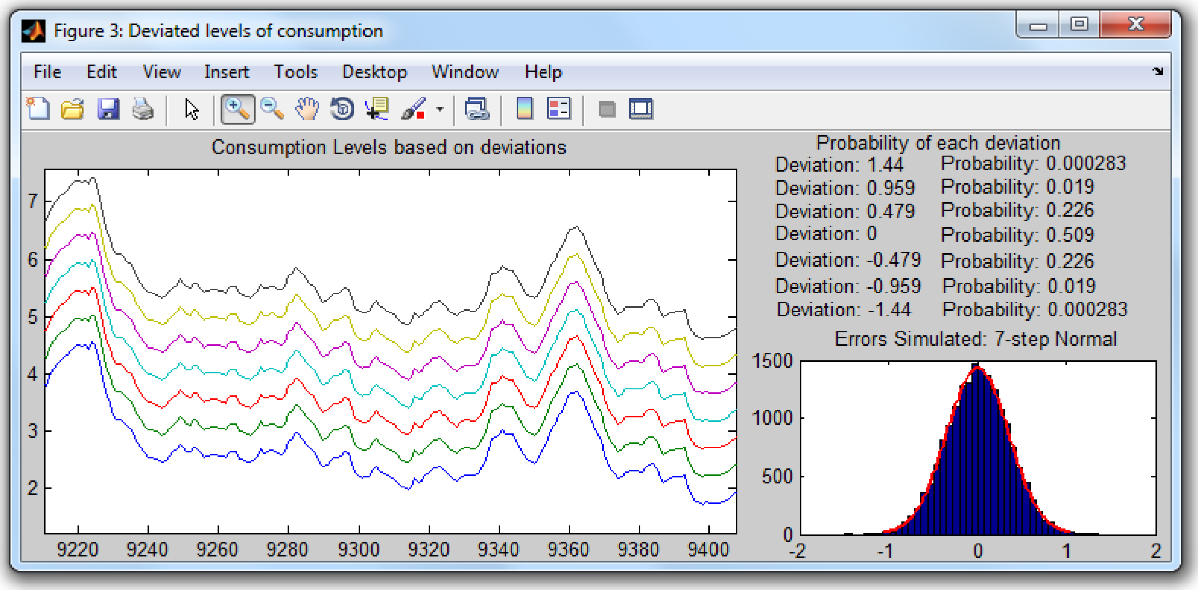


Figure 3: the seven deviated consumption levels obtained from the *seven-step approximation*. On top on the right, the levels of deviation with their historical probability; on the right bottom, the simulated errors distribution, same as the historical one, used in the following calculations.

Simulation

```
              Simulation validation
the whole year? "1" yes, "0" no: 0
select day to show: 27
Level chosen to start iterations (in %): 45
Temperature Forecast? "1" yes, "0" no: 1
```

Once the house is modelled, the *ToU* simulation's specifications are asked as input. Here it is selected to show the simulated *ToU* behaviour of the selected day (day 27). The starting storage level selected manually is 45%. Also the option of performing a temperature forecast is asked: it is selected *yes*.

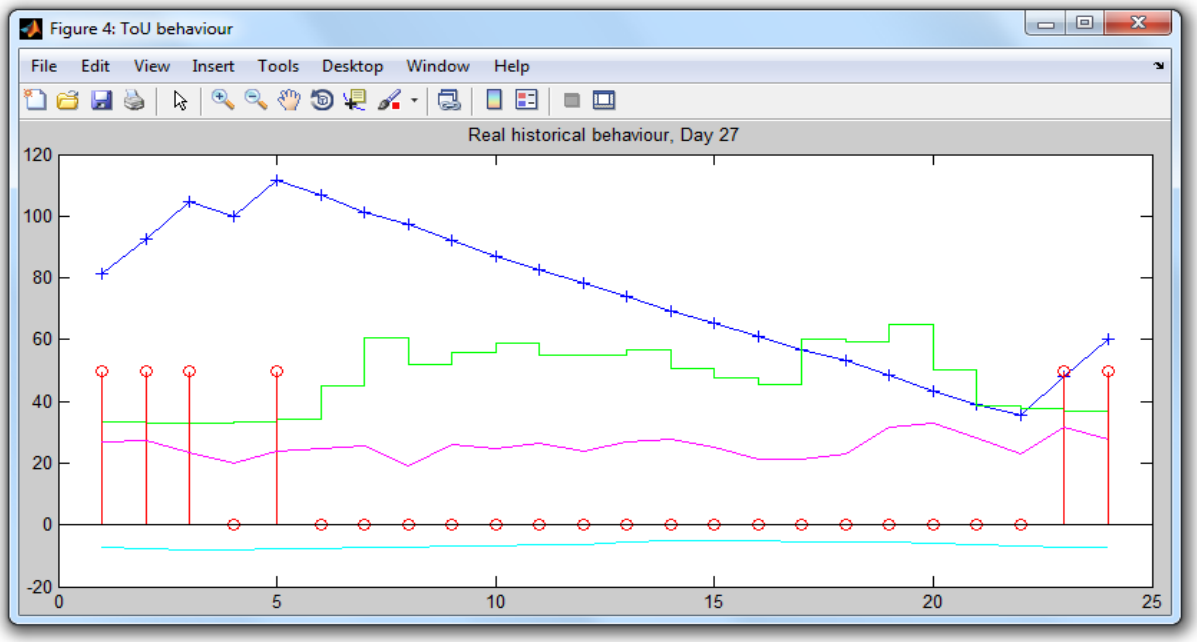


Figure 4: ToU simulation performed of how the customer would behave in case of not applying the control. Day selected by the controller: 338. In *green* it appears the Elspot Price, in *magenta* the simulated heating consumption (by using the forecasted temperature, shown in *cyan*), and in *dark blue* the blind storage evolution level.

Stochastic Genetic Algorithm's solution

```
----------------------------------------------------------
          End of STOCHASTIC GENETIC ALGORITHM
----------------------------------------------------------
Money Earned with this solution: 0.39083 €
Minimum charging hours needed: 4
Number of hours offered by this solution: 10
Storage level for tomorrow: 100%
Heat Consumption plan: 140 kWh
----------------------------------------------------------
```

Finally, the program gives the result obtained by the SGA. The program computes the costs of the control given; the formula used is:

$$Profit = Incomes - Costs \qquad [€] \qquad (39)$$

The incomes are computed as

$$\boldsymbol{Incomes} = P_B(kW) \cdot \frac{1MW}{1000\ kW} \cdot sum(n_{l.h.})(h) \cdot Tariff\ price\left(\frac{€}{MWh}\right) \qquad (40)$$

Every loading hour, the boiler works with its maximum power, meaning a load of $P_{boiler} \cdot 1h\ \ (kWh)$ every hour. By applying a single tariff price to the customer[45], all the loading hours are charged with the same price. $sum(n_{l.h.})$ represents the number of loading hours of this day.

The costs are computed as

$$\boldsymbol{Costs} = P_B(kW) \cdot \frac{1MW}{1000\ kW} \cdot sum(Elspot\ Prices\ of\ selected\ hours)\left(\frac{h \cdot €}{MWh}\right) \qquad (41)$$

Every hour of the Eslpot curve has a different energy price. For computing the costs, only the used hours' prices of energy are summed and translated into € with the amount of energy loaded ($P_{boiler} \cdot number\ of\ hours\ \ (kWh)$). The price tariff used is 38.88 €/MWh.[46]

[45] This single tariff price presented in two paragraphs.
[46] Average Esplot Price between 2012-2013.

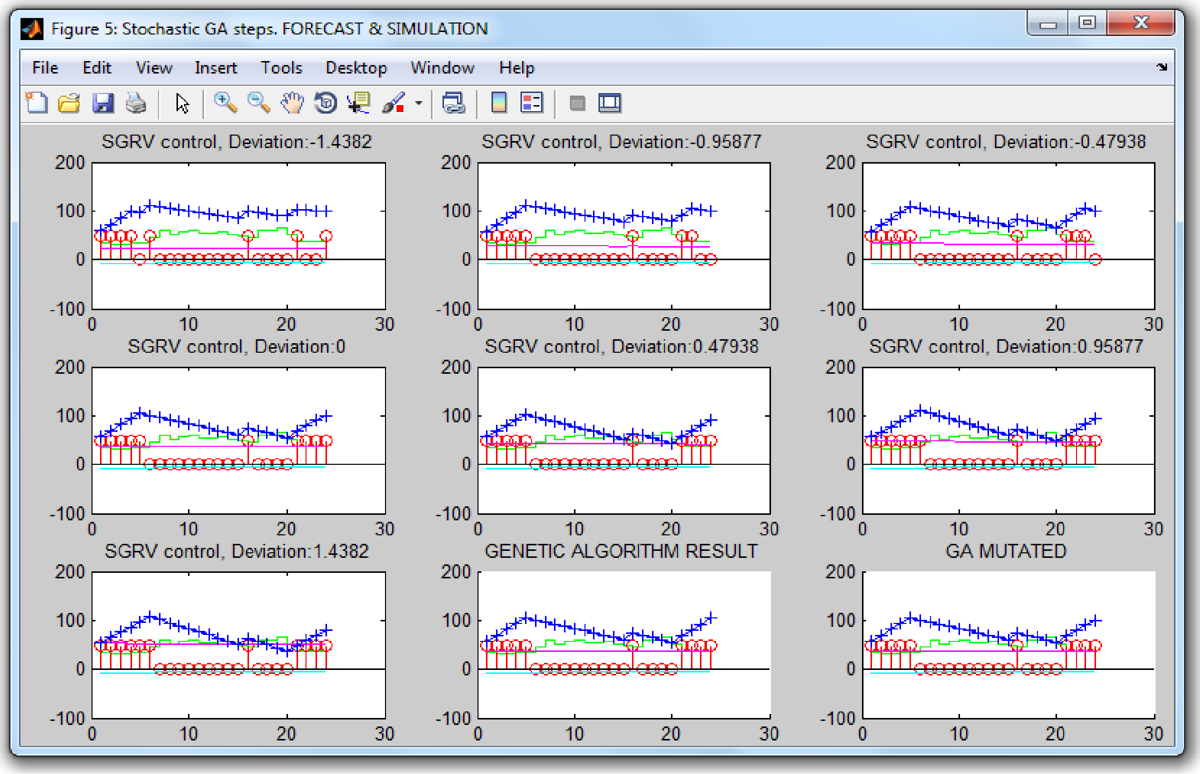
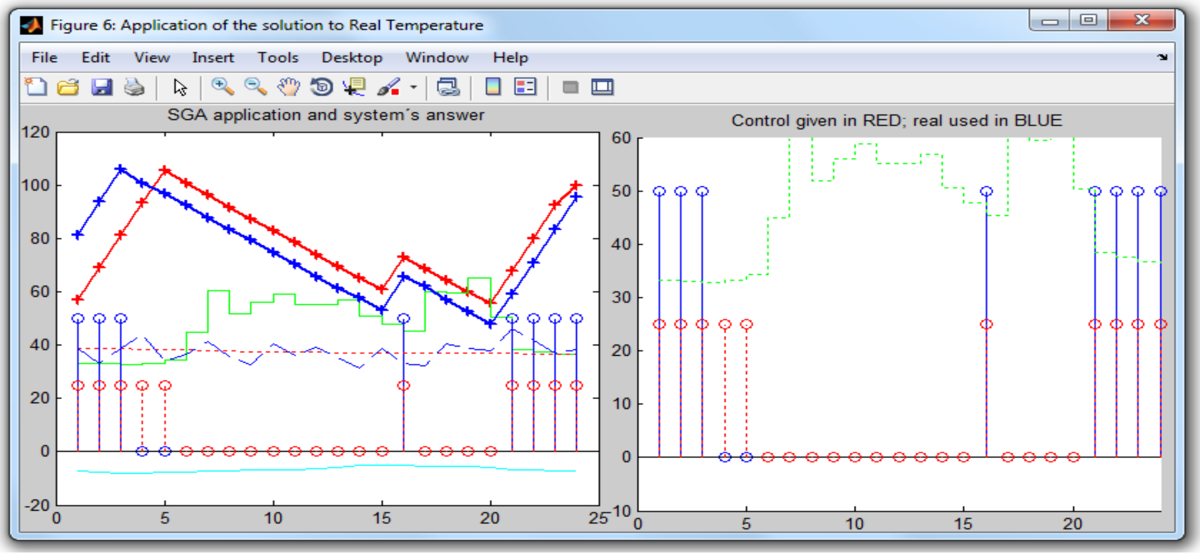


Figure 5: The solution of the *Stochastic Genetic Algorithm*. From the first to the seventh picture it is shown the control applied to the *Second Generation Response Vectors*[47]; the picture called *Genetic Algorithm's Result* is the solution obtained with the standard GA's application[48] and the last picture, called *GA Mutated* presents the final response vector given to the customer.

The SGA solution is accepted by the controller. It can be noticed that the SGA avoids charging when the Elspot price is high. Because of the picture size, it may not be appreciated that the solution selects the cheap hours of the day to perform the answer; a bigger picture can be found on *Figure 6*, shown on the next page. It is presented in this way in *Figure 5* to show the similarities between all the SGRV and the final offspring vector selected.



[47] See Chapter 5.
[48] See the block diagram, Chapter 5, page 33.

Figure 6: In this picture it is shown in *red* the SGA's response vector and the storage level evolution estimated, and in *blue* it is presented the one used by the client. The controller offered two loading hours at 4 am and 5 am that the customer did not use. This information is recorded.

Real consumption after control.

```
------------------------------------------------------------
------------------ END OF THE DAY -----------------------
Number of Real Loading Hours: 8
Final estimation of Storage Level for Tomorrow: 100%
Economic balance from real consumption: 0.24664 €
---------------------------------------------------------
>>
```

At the end of the day, the real balance is done. Here it can be seen on *Economic balance from real consumption* the profit or losses obtained during this day once the customer has used the control given. In this case, the economic balance is positive (0.24664 €).

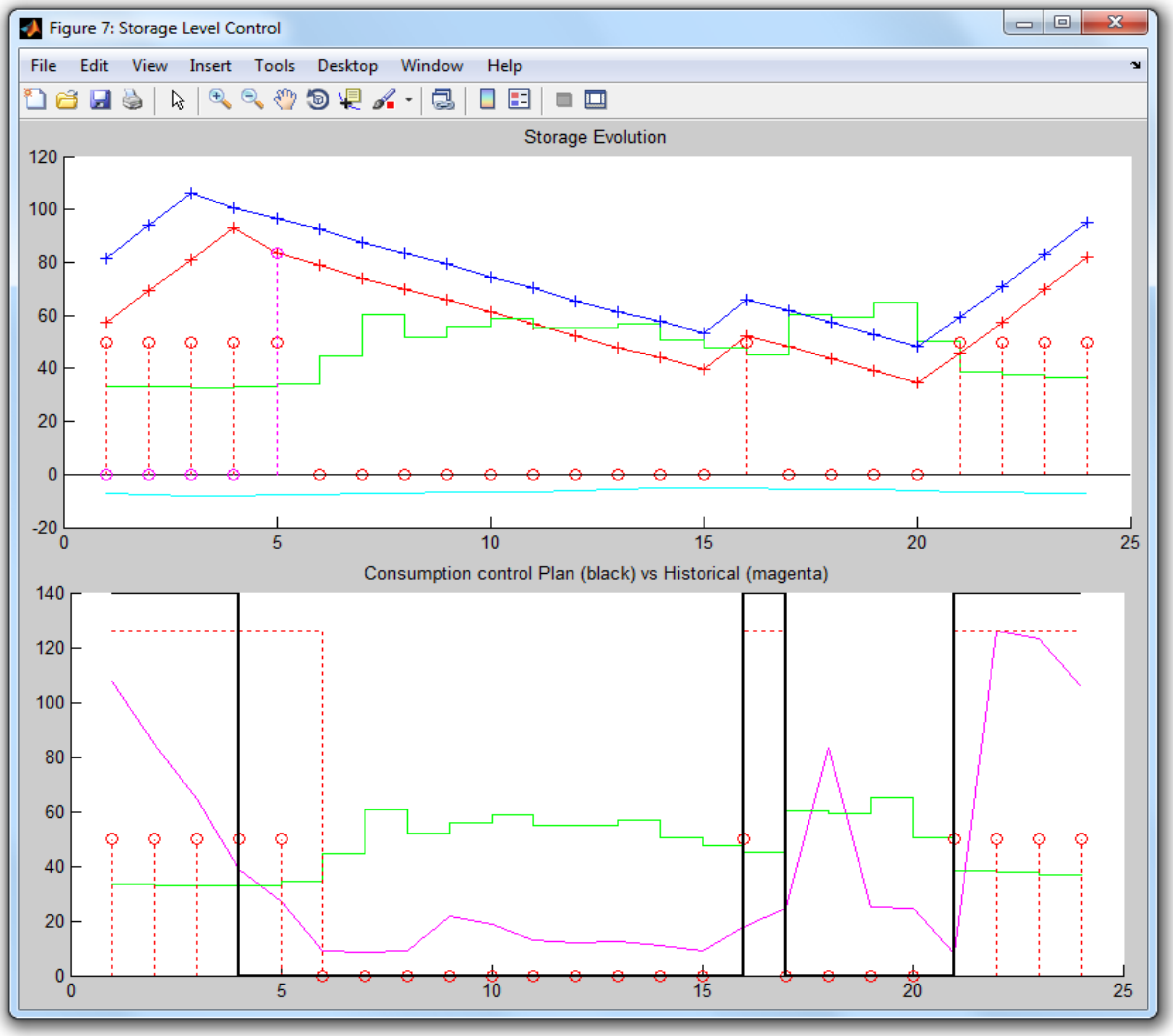


Figure 7: Storage level control and consumption plan. In *magenta* it is shown the historical consumption record, whereas in *black* it is presented the consumption plan for loading the storage. Both are presented in real scale. The *consumption plan* keeps the boiler working at full capacity $P_B = 14\ kW$ (the boiler load and historical consumption are plotted at 10:1 scale).

Control for the next day:

The control checking continues on the following days. For the next day in particular:

```
Command Window
>> Simulation
Initiate Control? "1" yes": 0
Reckoned level for tomorrow: 93.9554
____________________________________________
            Day after Control
____________________________________________
Temperature Forecast? "1" yes, "0" no: 1
```

Here, when the program asks "*Initiate Control?*", the given input is "*no*". The recorded "*Reckoned level for tomorrow"* obtained from the previous estimation is shown on screen.

```
----------------------------------------------------------
         End of STOCHASTIC GENETIC ALGORITHM
----------------------------------------------------------
Money Earned with this solution: 0.20858 €
Minimum charging hours needed: 1
Number of hours offered by this solution: 4
Storage level for tomorrow: 50.9087%
Heat Consumption plan: 56 kWh
-----------------------------------------------------------
----------------- END OF THE DAY ----------------------
Number of Real Loading Hours: 4
Final estimation of Storage Level for Tomorrow: 50.9087%
Economic balance from real consumption: 0.20858 €
---------------------------------------------------------
fx >>
```

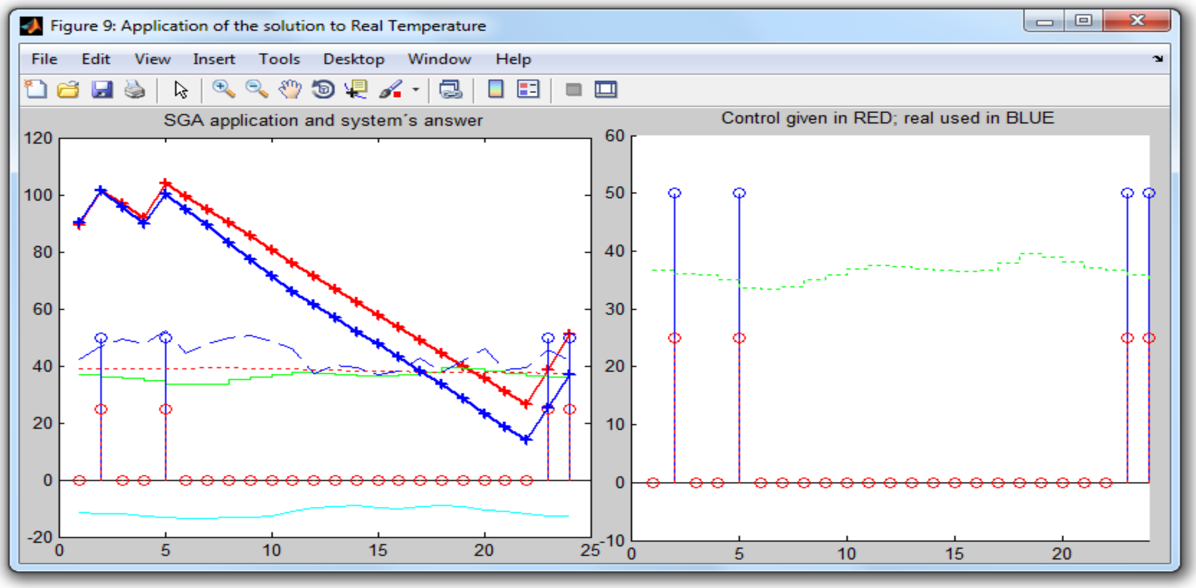


Figure 8: equivalent of figure 6 for this day. The program does not detect any variations.

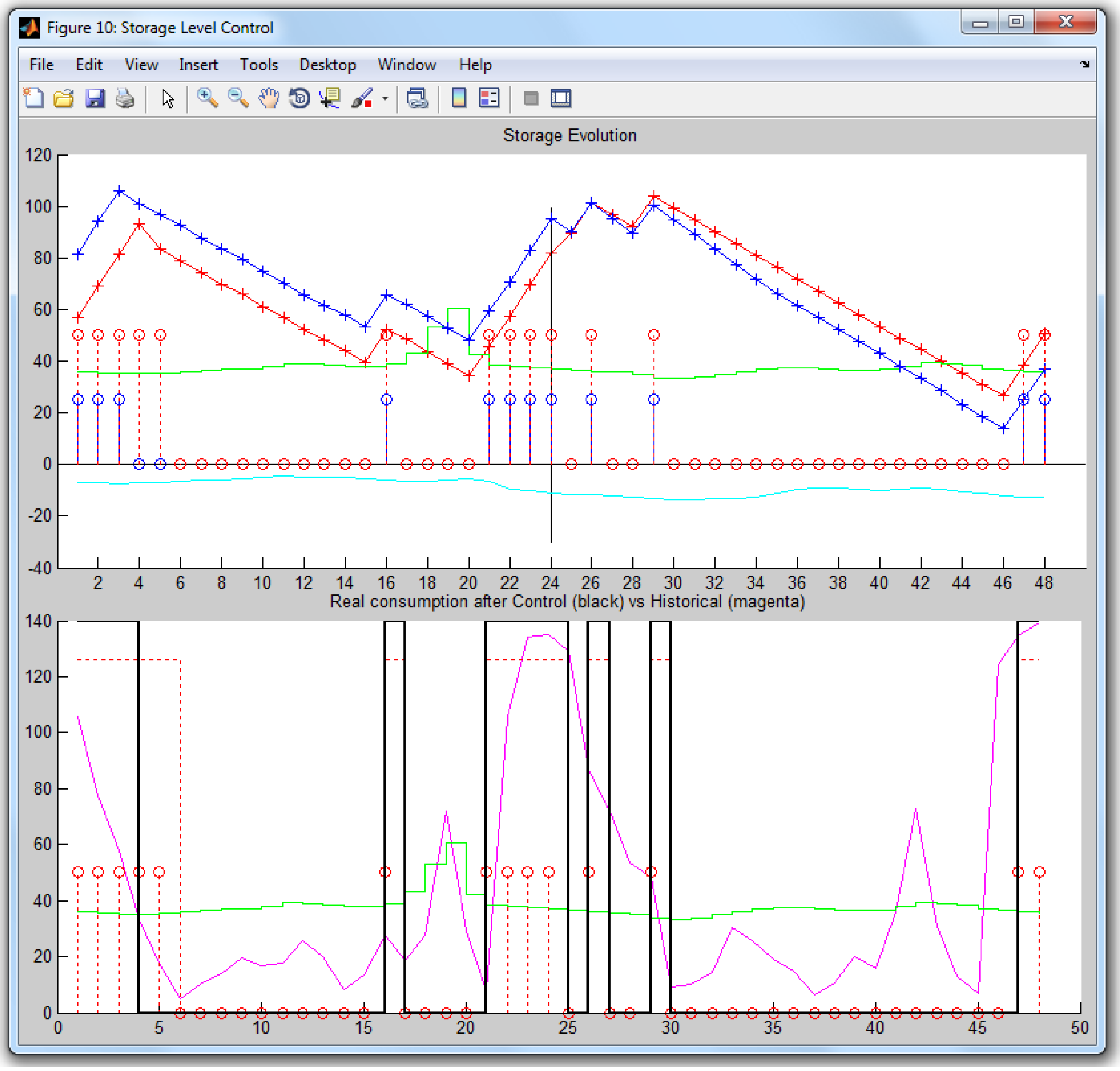
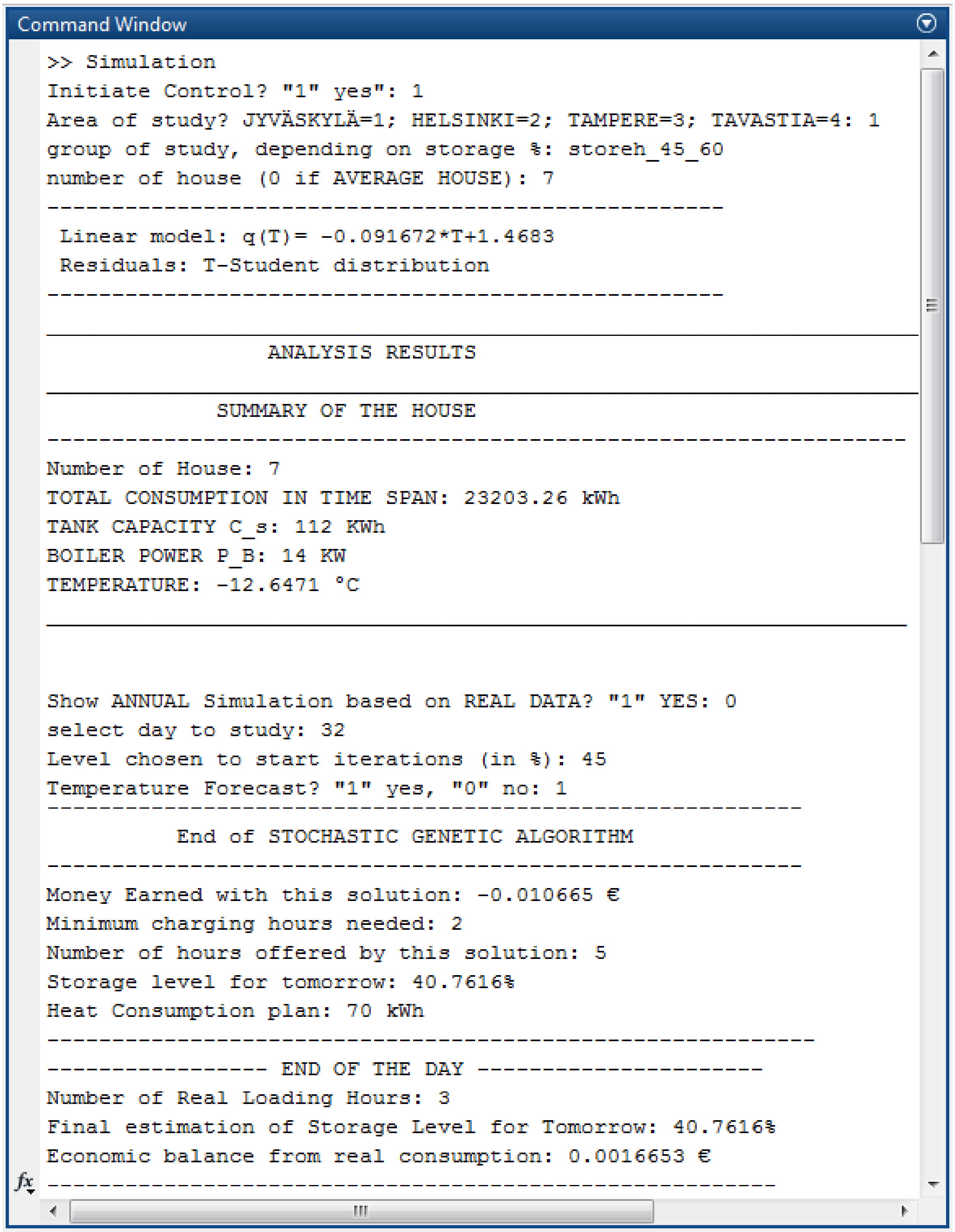


Figure 10: two-day control.

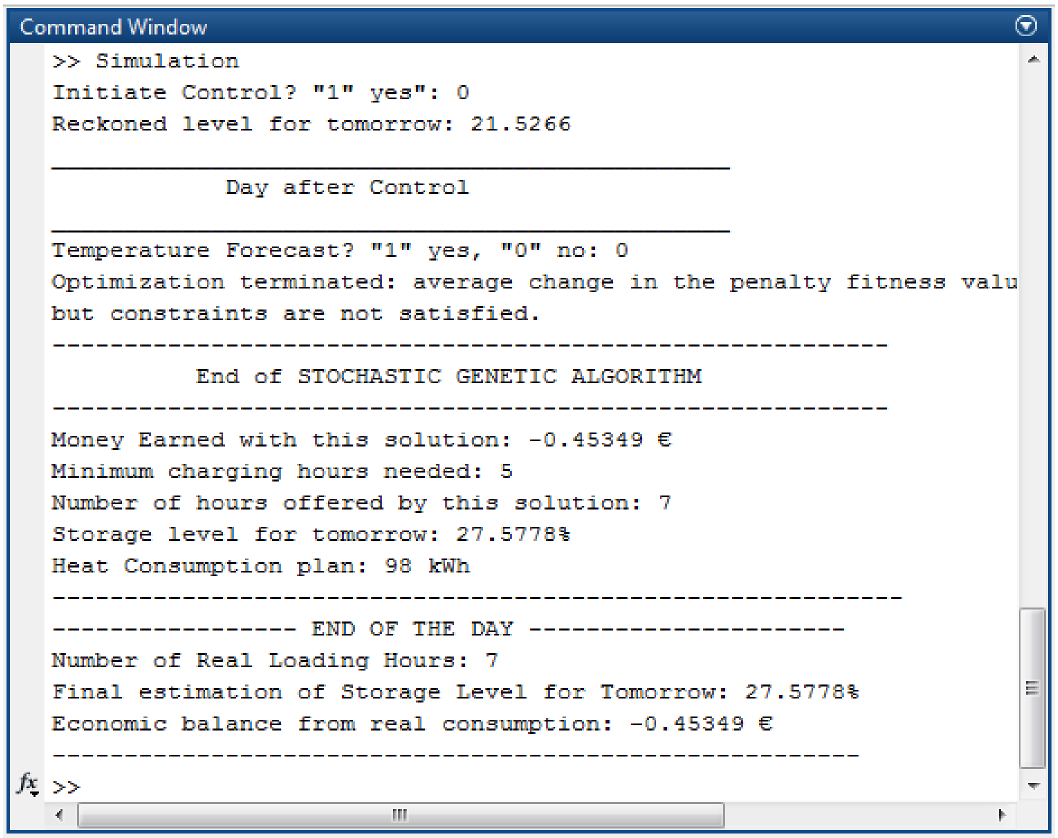
House 9, Tavastia



```
Command Window
>> Simulation
Initiate Control? "1" yes": 1
Area of study? JYVÄSKYLÄ=1; HELSINKI=2; TAMPERE=3; TAVASTIA=4: 1
group of study, depending on storage %: storeh_45_60
number of house (0 if AVERAGE HOUSE): 7
---------------------------------------------------
 Linear model: q(T)= -0.091672*T+1.4683
 Residuals: T-Student distribution
---------------------------------------------------

                 ANALYSIS RESULTS

             SUMMARY OF THE HOUSE
-----------------------------------------------------------------
Number of House: 7
TOTAL CONSUMPTION IN TIME SPAN: 23203.26 kWh
TANK CAPACITY C_s: 112 KWh
BOILER POWER P_B: 14 KW
TEMPERATURE: -12.6471 °C



Show ANNUAL Simulation based on REAL DATA? "1" YES: 0
select day to study: 32
Level chosen to start iterations (in %): 45
Temperature Forecast? "1" yes, "0" no: 1
---------------------------------------------------------
          End of STOCHASTIC GENETIC ALGORITHM
---------------------------------------------------------
Money Earned with this solution: -0.010665 €
Minimum charging hours needed: 2
Number of hours offered by this solution: 5
Storage level for tomorrow: 40.7616%
Heat Consumption plan: 70 kWh
----------------------------------------------------------
----------------- END OF THE DAY ----------------------
Number of Real Loading Hours: 3
Final estimation of Storage Level for Tomorrow: 40.7616%
Economic balance from real consumption: 0.0016653 €
fx -------------------------------------------------------
```

Figure 45 - Control to House 9, Tavastia, day 32. Positive economic balance.

The control is performed on day 32 (1st of January 2012, average Elspot price: $83.37\ €/MWh$) and the following day (day 33, average Elspot price: $101.26\ €/MWh$). These days exhibited very high prices. The selected house has a big storage capacity ($112\ kWh$). The control optimizes the loading period while minimizing economic losses.



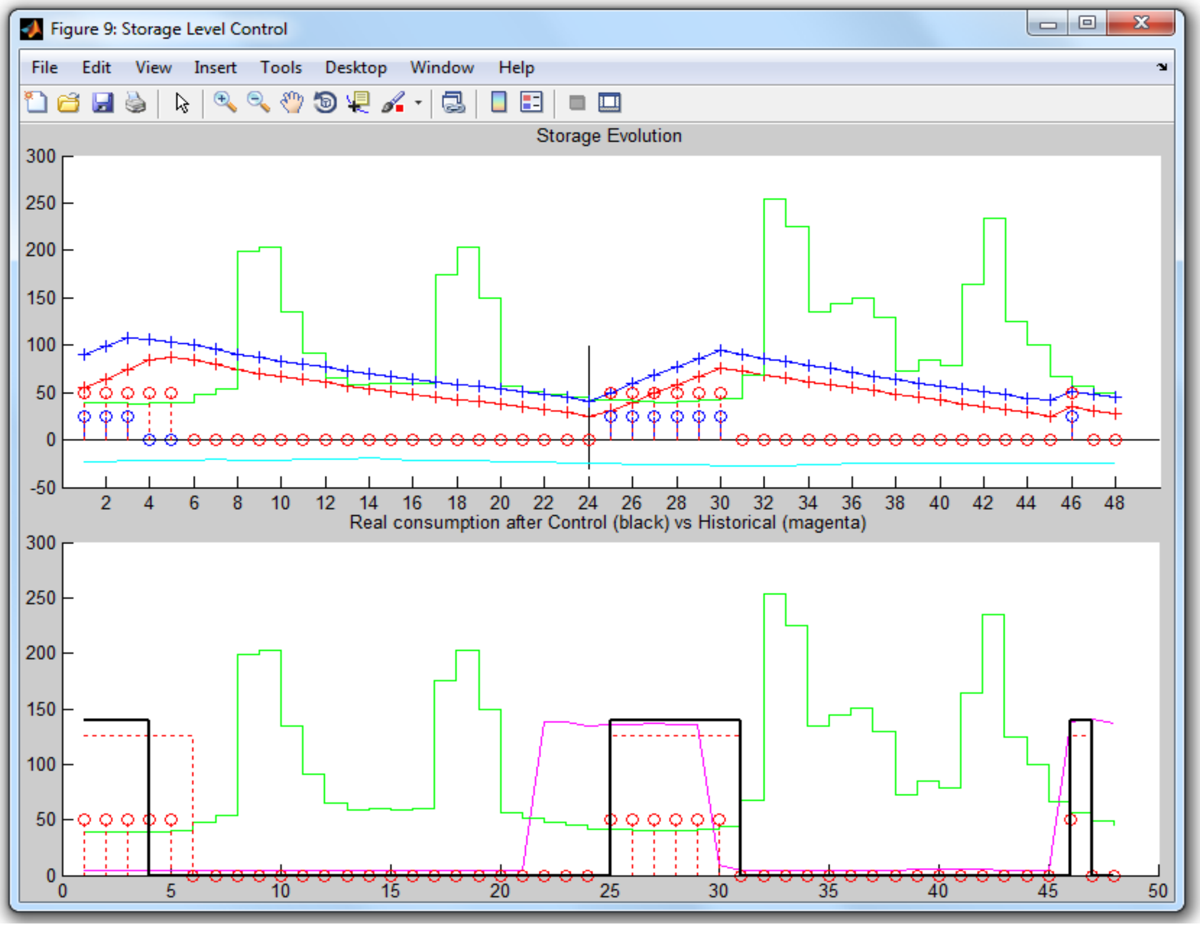


Figure 46 - Control to House 9, Tavastia, day 33. Negative economic balance, but the losses are minimum since they have been selected the cheapest hours to load.

Chapter 9. SUMMARY AND CONCLUSIONS

The purpose of this study was to perform a demand control of customers that exhibit a characteristic consumption pattern because of owning a TES system by estimating their storing level at the end of the day. This estimation allows the Distribution Company to track the customer in order to perform a daily consumption plan that fits his needs, while reducing unexpected consumption peaks that may cause high economic losses to the company.

Elspot is the day-ahead market of electrical energy in the Nordic countries. The distribution companies know the Elspot price evolution one day-ahead, and know which hours are profitable for them to buy electricity. These customers with TES are usually subscribed to *Time of Use* tariffs to load energy when it is cheaper to the distributor, but they often exhibit extra consumptions out of this *Time of Use* to cover their total demand. This extra consumption is *a priori* unpredictable and it appears regardless the Elspot price: reason why Distribution Companies search for ways to predict and control these consumptions.

Several tools have been presented in this work in order to control the demand. These tools work with the record of the electric meter as input data. The *cluster tool* and the *storage capacity selector tool* have been used to set the characteristic values and the energy storing capacity of each house; the linear approximation based on the regression model and the consumption deviations study have been set inside the *model of daily consumption*, that simulates the energy consumption from a temperature profile.

The key tool of this work is the *Stochastic Genetic Algorithm* (SGA), an optimization tool that uses the standard *Genetic Algorithm* (GA) to evolve and select a control vector among a population of possible solutions generated with pre-selected criteria; this starting population is unique to every customer because it is generated from his metered consumption levels. It is named "stochastic" because it introduces the randomness of the electric consumption as a creation criterion of the solution. The improvement of the SGA is to speed up the convergence of the standard GA to reach an optimal solution.

The demand forecast was performed one day ahead, and from it a consumption plan has been derived: the SGA's outcome set the plan that was later given to the customer as a vector. Electric meters give a measurement per hour; this structure has been kept to create a 24-component vector, where each component can be 1 or 0: the algorithm gives a 1 every hour selected to load and a 0 during hours that are not offered to load. Every consumption plan had its consequent response vector to give, and the customer fit his consumption to the plan according to its needs. Once the day was over it has been studied if the customer followed this given vector or if he introduced any variations on it. These variations in the given vector have been detected, and then have been interpreted by the *control tool* to update the customer's information in order to improve the control for the following day. The control worked as a variations detection tool between the plan given by the controller and the actual consumption. This has been presented with several households during different days.

The detection of a variation is followed by an update on the controller's information. In this work these updates were selected conservatively in order to ensure the estimation remained under the actual storage level. Thus the control plan would offer a number of extra loading hours to the customer in case of having failed on the estimation or in case of facing an actual demand higher than the forecasted one. The so called *control tool* has been used for two main purposes: to estimate the real storage level of the house and to perform a final economic balance. The storage level estimation has been presented in different situations and explained.

It has found that the most profitable plan is to keep a similar structure to *ToU* hours combined with a smart selection of the cheapest hours during the daytime. In this way the distributor can select which hours to offer and to maximize the profit. When the program detects a house with a big storing capacity, the control mainly offers night hours; whereas on houses where the storing capacity is small, the control performs more elaborated consumption plans combining daytime hours and night hours.

The control registered variations from the consumption plan when the storage level was reaching one of the bounds: according to the variation registered, the controller can know if the tank was depleted or full during that hour.

It is fundamental to highlight the limitations that some developed tools may present when estimating the characteristic values of each customer. These limitations are motivated by the lack of available information of the studied customers, whose electric record has been the only data available. Every customer's consumption profile is like a fingerprint, unique, and some of them are difficult to cluster into a consumption group and, what is more, they do not give satisfactory information to determine the installed heating system. Because of their simplicity, these estimation tools have excluded from having TES those who had exhibited specific consumption patterns in order to give a generic answer to every house. As an example, the *Storage Capacity Selector tool* could have failed on its capacity estimations; nevertheless these estimations have been considered as valid since there was no way to verify them with real samples. The same concession has been applied to the estimation of the *boiler power* (P_B): some divergences have been observed between the recorded consumption peaks at night and the estimated maximum boiler power in the control's final graph of several houses, whereas in many others the approximation has fitted the record accurately. In order to improve these estimation tools it is recommended to perform an application of a real time study to house whose storing capacity, boiler power and heating system characteristics are known beforehand.

The robustness of a 0-1 control relies on its simplicity; the outcome is a stairstep input to the boiler in order to operate at full capacity, and a 0 switches off the boiler. Nonetheless, in terms of further work it is strongly suggested to study the efficiency and profitability of implementing a control with different power input levels instead of working with full capacity.

It has been defended the goodness of the *Stochastic Genetic Algorithm* as a demand forecast tool as well as a daily cost minimization tool for the distribution company. Finally, it is suggested for further work to carry out a deeper economic analysis of the SGA's application in order to find out the most profitable energy tariff for these type of customers.

Bibliography

[1] Wikipedia, the free encyclopedia, "www.wikipedia.org," Wikimedia Foundation, [Online]. Available: http://en.wikipedia.org/wiki/Smart_meter.

[2] "www.americanhistory.si.edu," Smithsonian Insitute, [Online].
Available: http://americanhistory.si.edu/powering/transmit/trmain.htm.

[3] Smart Regions, "European Smart Metering Landscape Report," Österreichische Energieagentur – Austrian Energy Agency (AEA), Vienna, 2011.

[4] Statistics Finland, "Statistics Finland," 13 November 2013. [Online].
Available: http://www.stat.fi/til/asen/2012/asen_2012_2013-11-13_tau_001_en.html.

[5] S. V. a. J. T. Salla Annala, "Demand Response from Residential Customers' Perspective," in *2012 9th International Conference on the European Energy Market (EEM)*, Florence, 2012.

[6] P. Pippuri, "Heating mode choices of Finnish households and the Energy Paradox", Master's Thesis, Department of Economics, Aalto University, School of Economics, Helsinki, 2012.

[7] N. H. F. P. A. Arteconti, "Domestic demand-side management (DSM): Role of heat pumps and thermal enery storage (TES) systems," Elsevier, 2012.

[8] H. Z. Alemayehu Gebremedhin, "Seasonal Heat Storages in District Heating Systems," Dept. of Management and Engineering/Energy Systems, Linköping University, Linköping, Sweden, Linköping.

[9] "www.kaukora.fi," kaukora Oy, [Online].
Available: http://www.kaukora.fi/en/accumulators.

[10] "energy.gov," U.S. Department of Energy, 15 June 2012. [Online].
Available: http://energy.gov/energysaver/articles/storage-water-heaters.

[11] J. W. D. Chvala, "Technology Potential of Thermal Energy Storage (TES) Systems in Federal Facilities," U.S. Department of Energy, 2001.

[12] "www.elenia.com," Elenia Oy, [Online].
Available: http://www.elenia.com/en/business/electricity.

[13] "Wikipedia, the free encyclopedia," Wikimedia Foundation, 31 July 2014. [Online]. Available: http://en.wikipedia.org/wiki/Electricity_retailing.

[14] "Nord Pool Spot," [Online]. Available: http://www.nordpoolspot.com/.

[15] M. Ali, "The Role of Electric Space Heating in Demand Response", Master's Thesis, Department of Electrical Engineering, School of Electrical Engineering, Aalto University, Helsinki, Finland, Helsinki, 2012.

[16] J. S. Anderpont, “Thermal Energy Storage: Solutions for Demand Management,” Association of Energy Engineers (AEE), 2003, 100:3, 66-80.

[17] C. S. T. M. A. M. Fortunato B., “Simple Mathematical Model of a Thermal Storage with PCM,” in *2012 AASRI Conference of Power and Energy Systems*, Bari, 2012, Volume 2, 2012, p. 241-248.

[18] M. K. G. P. H. Pyeongchan Ihma, “Science Direct,” 8 August 2004. [Online].

Available: http://www.sciencedirect.com/science/article/pii/S0378778804000337.

[19] V. Vélez, “Design of a Control Strategy for Optimal Operation of Autonomous Distributed Generation System for Electricity and Heat,” Master's of Science Thesis, Faculty of Electrical Engineering, Mathematics and Computer Sciences, Faculty of Applied Sciences, Delf University of Technology, Delf, 2010.

[20] C. A. Cruickshank, “Evaluation of a Stratified Multi-Tank Thermal Storage for Solar Heating Applications,” Bacherlor's Thesis, Departmnent of Mechanincal and Materials Engineering, Queen's University, Kingston, Ontario, Canada, 2009.

[21] A. G. C. Á. J. F. E. G. A. Molina, “Electrical Thermal Storage Modeling: a Tool to Evaluate New Opportunities and Bids for Residential Users in a Deregulated Market,” in *2003 IEEE Bologna PowerTech Conference, June 23-26*, Bologna, Italy, 2003.

[22] N. L. Shu-Xia Yang, “Power Demand Forecast based on Optimized Neural Networks by improved Genetic Algorithm,” in *Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006*.

[23] O. R. H. Alamos, “Genetic Algorithm model to control peak demand to feder capacity investment,” in *Power and Energy Society General Meeting, 2012 IEEE*, San Diego, CA, 2012.

[24] M. C. Sun-Jin Kim, “A Genetic Algorithm for Server Location and Storage Allocation in Multimedia-on-Demand Network,” School of Business, Information and Communications University58-4, Hwaam-Dong, Yuseong-Gu, Daejeon, Republic of Korea, Mobile Future and Symposium on Trends in Communications, 2003. SympoTIC '03. Joint First Workshop on, 26-28 October 2003.

[25] A. G. J. L. R. F. A. J. J. Cárdenas, “A Genetic Algorithm Approach to Optimization of Power Peaks in an Automated Warehouse,” Motion Control and Industrial Applications Group, Technical University of Catalonia, Terrassa, Catalonia, Spain, Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE, 3-5 November 2009, p. 3297 - 3302

[26] A. R. A. S. a. F. M. F. M. Enrico De Santis, “Genetic Optimization of a Fuzzy Control System for Energy Flow Management in Micro-Grids,” Information Engineering, Electronics and Telecommunications Dept., University of Rome La Sapienza, Rome, Italy, IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint, 24-28 June 2013, p.418-423.

[27] “Kaukora,” [Online]. Available:

http://www.kaukora.fi/sites/default/files/kaukorafiles/EN_brochures/Jaspi_TW_PW_brochure_EN .pdf.

[28] “Trianco,” [Online].

Available: http://www.trianco.co.uk/products/aztec-electric-boilers.

[29] "Relek," [Online]. Available: http://www.relek.se/elpanna.html.

[30] "Electric-Heating Company," [Online]. Available: http://www.electric-heatingcompany.co.uk/wp-content/uploads/2013/12/EHC-A6-Handbook-web.pdf.

[31] "Sillicon Valley Power, City of Santa Clara," [Online]. Available: http://siliconvalleypower.com/index.aspx?page=1922.

[32] "Daft Logic," [Online]. Available: http://www.daftlogic.com/information-appliance-power-consumption.htm.

[33] "wiseGEEK," [Online]. Available: http://www.wisegeek.com/what-countries-consume-the-most-electricity.htm.

[34] "Wikipedia, the free encyclopedia," Wikimedia Foundation, [Online]. Available: http://en.wikipedia.org/wiki/Mendelian_inheritance.

[35] M. Melanie, An Introduction to Genetic Algorithms, Cambridge, Massachusetts, London, England: Massachusetts Institute of Technology, MIT Press, 1999, p.2-24

[36] "www.statistics.com," [Online]. Available: http://www.statistics.com/glossary&term_id=710.

[37] "Wikipedia, the free enciclopedia," [Online].

[40] "Network Tariffs," Elenia Oy, 2013.

[41] "boilersheaters.com," New Home Energy, [Online]. Available: http://boilersheaters.com/compare-table-coefficient-of-performance-cop/.

[42] "www.stafor.lv," Stafor, Kvalitates Zime, 2009. [Online]. Available: http://www.stafor.lv/ion/index.php?part=2&part2=55&lng=en.

ANNEXES

1) Cluster Tool

Cluster the houses into different size's groups and provides information to the controller about the population of customers.

Contents

FUNCTION FOR COMPUTING THE ANNUAL CONSUMPTION OF EACH HOUSE

```
% Instructions to make Inputs:
% Inputs should follow some rules, such as:
%              Number of lines = Number of houses;
%              Number of colums = Number of hours metered
%-------------------------------------------------------------------------
function [storeh_45_60 storeh_60_75 storeh_75_100]=Cluster_Tool(name)
```

```
num=length(name(:,1));
long=length(name(1,:));

%& Annual consumption for each household
for k=1:num
          v_areas(k)=trapz(name(k,:));
end

%& Consumption day by day (from 22 to 46 hours THE FIRST)

Y=1:24:long+1;                 %divide one year in days
houses=1:1:num;
start_time=22;
end_time=7;

for j=1:length(houses)
    for i=1:length(Y)-2

% Integrate of the whole day from 22:00 to 46:00 on the first round
    v_daily_cons(i)=trapz(name(j,Y(i)+start_time-1:Y(i+1)+start_time-1));
    end;
daily_cons(j,:)=v_daily_cons;
end
```

Stored capacity

```
for j=1:length(houses)
    for i=1:length(Y)-2
% Integrate ToU hours from 22:00 to 7:00 of the first day
        v_daily_heat(i)=trapz(name(j,Y(i)+start_time-1:Y(i+1)+end_time-1));
    end;
daily_heat(j,:)=v_daily_heat;
end
```

Percentaje of Daily Energy Stored out of Daily Consumption

```
for i=1:num                    %all the houses
    for j=1:length(v_daily_heat)                %daily in one year
        if daily_heat(i,j)>0
        Heat_per100(i,j)=daily_heat(i,j)/daily_cons(i,j);
        else Heat_per100(i,j)=0;
        end
    end
    night_cons100(i)=mean(Heat_per100(i,:));
end
```

STORAGE or NO-STORAGE

```
sto=1;
no_sto=1;
for i=1:num               %line1: position of the house.
    if night_cons100(i)>0.45
    STORAGE(1,sto)=i;
    sto=sto+1;
    else
    NO_STORAGE(1,no_sto)=i;
    no_sto=no_sto+1;
    end
end
for i=1:length(STORAGE(1,:))
    for j=1:length(name(1,:))
     storeh_ON(i,j)=name(STORAGE(1,i),j);
    end
end
```

CREATING THE OUTPUT OF THE PROGRAM

```
for i=1:length(STORAGE)    %HOUSES WITH STORAGE
    STORAGE(2,i)=v_areas(STORAGE(1,i));      %line2: total anual consumption
    STORAGE(3,i)=night_cons100(STORAGE(1,i));  %line3: %Heat
    STORAGE(4,i)=STORAGE(2,i)*STORAGE(3,i);    %line4: Annual Heat
end
for i=1:length(NO_STORAGE)    %HOUSES WITH STORAGE
    NO_STORAGE(2,i)=v_areas(NO_STORAGE(1,i));      %line2: total anual consumption
    NO_STORAGE(3,i)=night_cons100(NO_STORAGE(1,i));  %line3: %Heat
    NO_STORAGE(4,i)=NO_STORAGE(2,i)*NO_STORAGE(3,i);    %line4: Annual Heat
end
```

Generating the function's output

```
i1=0; i2=0; i3=0;
stores=length(STORAGE(1,:));
for i=1:stores
    if STORAGE(3,i)<=0.6
        i1=i1+1;
        STOR_45_60(:,i1)=STORAGE(:,i);
        X=STORAGE(1,i);
        vec=name(X,:);
        storeh_45_60(i1,:)=vec;
    elseif (STORAGE(3,i)<=0.75)&(STORAGE(3,i)>0.6)
        i2=i2+1;
        STOR_60_75(:,i2)=STORAGE(:,i);
        X=STORAGE(1,i);
        vec=name(X,:);
        storeh_60_75(i2,:)=vec;
    elseif (STORAGE(3,i)>0.75)&(STORAGE(3,i)<=1)
        i3=i3+1;
        STOR_75_100(:,i3)=STORAGE(:,i);
        X=STORAGE(1,i);
        vec=name(X,:);
        storeh_75_100(i3,:)=vec;
    end
end
```

2) LEGO Function:

Data study to obtain the storage capacity of each house. Selection of winter months for the purpose, as they are the most significant when studying heating systems. Output: summary of the house's characteristics.

Contents

LEGO function

Once we have the houses divided into groups, LEGO_Simulation studies each single house in order to get the Temperature reference and the Capacity.

```
function [Capacity Boiler_Power Temp_Limit Annual_Cons WinterTime Selection]=LEGO_Simulation(Average_house,num,Temp,det);
```

Daily consumption (from 22 to 46 hours THE FIRST)

```
Annual_Cons.length=length(Average_house(1,:));

Y=1:24:Annual_Cons.length+1;
               % divide one year in days
start_time=22;
end_time=7;
for i=1:length(Y)-2
    Annual_Cons.v_daily_cons(i)=trapz(Average_house(1,Y(i)+start_time-1:Y(i+1)+start_time-1));
    % integrate of the whole day from 22:00 to 46:00 on the first round
    % the first value of the string is the consumption 01.01.2012-22:00 to
    % 02.01.2012-7:00  TOMORROW
end;
```

Consumption in ToU & Not_ToU:

```
for i=1:length(Y)-2
    Annual_Cons.v_heat(i)=trapz(Average_house(1,Y(i)+start_time-1:Y(i+1)+end_time-1)); %integrate ToU hours
    Annual_Cons.v_daytime(i)=Annual_Cons.v_daily_cons(i)-Annual_Cons.v_heat(i);
    % from 22:00 to 7:00 of the first day
end;
```

Daily Temperature

```
for i=1:length(Y)-2
     Annual_Cons.Average_Temp(i,1)=sum(Temp(Y(i):Y(i+1),2))/24;  %Tem
end;
```

Average daily consumption

```
long_d=length(Annual_Cons.v_daytime);
Annual_Cons.Average_daytime=sum(Annual_Cons.v_daytime)/long_d;

Annual_Cons.max_heat=max(Annual_Cons.v_heat)/9;
Annual_Cons.min_daytime=mean(Annual_Cons.v_daytime)/15;

for i=1:length(Y)-2
    Annual_Cons.Heat_per100(i)=Annual_Cons.v_heat(i)/Annual_Cons.v_daily_cons(i);
end
```

WINTER TIME

```
WinterTime.winter1.days=Average_house(1,1:1608);        % from 01.01.2012-00:00 to 07.03.2012-23:00
WinterTime.winter2.days=Average_house(1,7993:10992);    % from 29.11.2012-00:00 to 02.04.2013-23:00
WinterTime.winter3.days=Average_house(1,16051:17568);
% In these way complete days are used
WinterTime.winter1.Temp=Temp(1:1608,2);        % from 01.01.2012-00:00 to 07.03.2012-23:00
WinterTime.winter2.Temp=Temp(7993:10992,2);    % from 29.11.2012-00:00 to 02.04.2013-23:00
WinterTime.winter3.Temp=Temp(16051:17568,2);
%-----------------------------------------
% Winter1
long1=length(WinterTime.winter1.days);
k=1:24:long1+1;                 % divide one year in days
for i=1:length(k)-2
    WinterTime.winter1.day_cons(i)=trapz(WinterTime.winter1.days(k(i)+start_time-1:k(i+1)+start_time-1));
    % integrate of the whole day from 22:00 to 46:00 on the first round
end;
for i=1:length(k)-2
    WinterTime.winter1.heat(i)=trapz(WinterTime.winter1.days(k(i)+start_time-1:k(i+1)+end_time-1));
    % integrate ToU hours
    WinterTime.winter1.daytime(i)=WinterTime.winter1.day_cons(i)-WinterTime.winter1.heat(i);
    % from 22:00 to 7:00 of the first day
end;
long_d=length(WinterTime.winter1.daytime);
WinterTime.winter1.av_cons_daytime=sum(WinterTime.winter1.daytime)/long_d;
[val pos]=max(WinterTime.winter1.heat);     % 9 hours between 22:00 and 7:00
WinterTime.winter1.max_heat_cons.position=pos;
WinterTime.winter1.max_heat_cons.value=val/9;
WinterTime.winter1.average_day_cons=mean(WinterTime.winter1.daytime)/15;  % 15 hours between 7:00 and 22:00

for i=1:length(WinterTime.winter1.daytime)
   WinterTime.winter1.percentajes(i)=WinterTime.winter1.heat(i)/WinterTime.winter1.daytime(i);
end
%----------------------------------------
% Winter2
long2=length(WinterTime.winter2.days);
k=1:24:long2+1;                 % divide one year in days
for i=1:length(k)-2
    WinterTime.winter2.day_cons(i)=trapz(WinterTime.winter2.days(k(i)+start_time-1:k(i+1)+start_time-1));
    % integrate of the whole day from 22:00 to 46:00 on the first round
end;
for i=1:length(k)-2
    WinterTime.winter2.heat(i)=trapz(WinterTime.winter2.days(k(i)+start_time-1:k(i+1)+end_time-1));
    % integrate ToU hours
    WinterTime.winter2.daytime(i)=WinterTime.winter2.day_cons(i)-WinterTime.winter2.heat(i);
    % from 22:00 to 7:00 of the first day
end;
long_d=length(WinterTime.winter2.daytime);
WinterTime.winter2.av_cons_daytime=sum(WinterTime.winter2.daytime)/long_d;
[val pos]=max(WinterTime.winter2.heat);     % 9 hours between 22:00 and 7:00
WinterTime.winter2.max_heat_cons.position=pos;
WinterTime.winter2.max_heat_cons.value=val/9;
WinterTime.winter2.average_day_cons=mean(WinterTime.winter2.daytime)/15;   % 15 hours between 7:00 and 22:00

for i=1:length(WinterTime.winter2.daytime)
    WinterTime.winter2.percentajes(i)=WinterTime.winter2.heat(i)/WinterTime.winter2.daytime(i);
end
```

```
%---------------------------------------
% Winter3
long3=length(WinterTime.winter3.days);
k=1:24:long3+1;                    % divide one year in days
for i=1:length(k)-2
    WinterTime.winter3.day_cons(i)=trapz(WinterTime.winter3.days(k(i)+start_time-1:k(i+1)+start_time-1));
    % integrate of the whole day from 22:00 to 46:00 on the first round
end;
for i=1:length(k)-2
    WinterTime.winter3.heat(i)=trapz(WinterTime.winter3.days(k(i)+start_time-1:k(i+1)+end_time-1));
    % integrate ToU hours
    WinterTime.winter3.daytime(i)=WinterTime.winter3.day_cons(i)-WinterTime.winter3.heat(i);
    % from 22:00 to 7:00 of the first day
end;
long_d=length(WinterTime.winter3.daytime);
WinterTime.winter3.av_cons_daytime=sum(WinterTime.winter3.daytime)/long_d;
[val pos]=max(WinterTime.winter3.heat);     % 9 hours between 22:00 and 7:00
WinterTime.winter3.max_heat_cons.position=pos;
WinterTime.winter3.max_heat_cons.value=val/9;
WinterTime.winter3.average_day_cons=mean(WinterTime.winter3.daytime)/15;    % 15 hours between 7:00 and 22:00

for i=1:length(WinterTime.winter3.daytime)
    WinterTime.winter3.percentajes(i)=WinterTime.winter3.heat(i)/WinterTime.winter3.daytime(i);
end

WinterTime.Selection.maximo=max(WinterTime.winter1.max_heat_cons.value,WinterTime.winter2.max_heat_cons.value);
% Set upper bound
WinterTime.Selection.maximo=max(WinterTime.Selection.maximo,WinterTime.winter3.max_heat_cons.value);
WinterTime.Selection.minimo=max(WinterTime.winter1.average_day_cons,WinterTime.winter2.average_day_cons);
% Set lower bound
WinterTime.Selection.minimo=min(WinterTime.Selection.minimo,WinterTime.winter3.average_day_cons);
WinterTime.Selection.average_daytime=(WinterTime.winter1.av_cons_daytime+WinterTime.winter2.av_cons_daytime...
    +WinterTime.winter3.av_cons_daytime)/3;
% 100 of sto capacity + 1.1*daily average cons
WinterTime.Selection.Limit=WinterTime.Selection.maximo*9+1.1*WinterTime.Selection.minimo*(22-7);
```

Display

```
disp('_____________________________________________________________________')
disp('                    ANALYSIS RESULTS                              ')
disp('_____________________________________________________________________')

perref=0.85;
maximo=WinterTime.Selection.maximo;

Selection=Day_Selector(Annual_Cons,WinterTime,perref,maximo);
while Selection.Temperature_Limit==50;
  perref=perref-0.05;
  maximo=0.95*maximo;
  WinterTime.Selection.maximo=maximo;
  Selection=Day_Selector(Annual_Cons,WinterTime,perref,maximo);
end
```

Boiler Power and Storage Capacity

```
WinterTime.Winter_days_all=[WinterTime.winter1.days WinterTime.winter2.days WinterTime.winter3.days];
long=length(WinterTime.Winter_days_all);
Y_winter=1:24:long;
for i=1:length(Y_winter)-1
     max_Power(i)=floor(max(WinterTime.Winter_days_all(1,Y_winter(i):Y_winter(i+1))));
end
z=unique(max_Power);
[X]=hist(max_Power,length(z));
val_power=length(X);
for i=1:length(X)
    if X(i)>30 Boiler_Power=z(i);
    end
end
```

```
h_val=Average_house(1,Y(Selection.Day_Limit)+start_time-2:Y(Selection.Day_Limit+1)+start_time-2);
for i=1:25
    if h_val(i)<=0.6*Boiler_Power Boiler_ON(i)=0;
    else Boiler_ON(i)=1;
    end
end
Selection.position=1;
Switch=Boiler_ON(Selection.position);
ON_hours=0;
while Switch==0
    Selection.position=Selection.position+1;
    Switch=Boiler_ON(Selection.position+1);
end
start_load=Selection.position;
Switch=1;
while Switch==1
    Switch=Boiler_ON(Selection.position+1);
    Selection.position=Selection.position+1;
end
end_load=Selection.position;
Selection.charging_hours=end_load-start_load;
Capacity=Boiler_Power*Selection.charging_hours;
```

```
disp('             SUMMARY OF THE HOUSE')
disp('-----------------------------------------------------------------')
X=['Number of House: ',num2str(num)];
disp(X)
X=['TOTAL CONSUMPTION IN TIME SPAN: ',num2str(sum(Average_house(1,:))),' kWh'];
disp(X)
X1=['TANK CAPACITY C_s: ',num2str(Capacity),' KWh'];
disp(X1)
X1=['BOILER POWER P_B: ',num2str(Boiler_Power),' KW'];
disp(X1)
if Selection.nol==0 Temp_Limit=Selection.Temperature_Limit*0.7+Annual_Cons.Average_Temp(Selection.Day_Limit-1)*0.3;
else Temp_Limit=Selection.Temperature_Limit;
end
if Selection.nol==1 disp('Maximum Storage Capacity not reached')
end
if Selection.nol==2 disp('Strange Case, can cause error')
end
X3=['TEMPERATURE: ',num2str(Temp_Limit),' °C'];
disp(X3)
disp('_________________________________________________________________')
disp('   ')
disp('   ');
```

```
end
```

3) Day Selector

```
function Selection=Day_Selector(Annual_Cons,WinterTime,perref,maximo)
Y=1:24:Annual_Cons.length+1;
Selection.Temperature_ref=10;
Selection.Position=0;
Selection.Per100_Reference=perref;
talvi=0;
Selection.Temperature_Limit=50;
Selection.Day_Limit=1;
for i=1:length(Y)-2
    Selection.Position=Selection.Position+1;
    if (Annual_Cons.v_daily_cons(i)>=(0.9*maximo*9))&...
            (Annual_Cons.Heat_per100(i)>Selection.Per100_Reference)
         % Isolating days of great consumption at night the consumption of
         % the whole day is greater than the max storage capacity.
         % Big consumption, big storage percentaje
            if (Annual_Cons.v_daily_cons(i)<=WinterTime.Selection.Limit)
                % Case STO lim NOT REACHED yet
                  if Annual_Cons.v_daytime(i)<1.1*Annual_Cons.Average_daytime
                      % NO EXTRA HEAT in daytime
                        if (Annual_Cons.Average_Temp(i)<Selection.Temperature_ref)
                           % lowest of low temperatures
                           Selection.Temperature_Limit=Annual_Cons.Average_Temp(i);
                           Selection.Day_Limit=i;
                           Selection.Per100_Reference=Annual_Cons.Heat_per100(i);
                           Selection.Temperature_ref=Selection.Temperature_Limit;
                           Selection.bandera=1;
                           Selection.nol=0;
                        else Selection.bandera=4;
                        end
                  else Selection.bandera=3;
                  end
            else  Selection.bandera=2;                   % case LIMIT NOT REACHED
            end
    end
end
[valm posm]=min(Annual_Cons.Average_Temp);
if Selection.Temperature_ref==valm
    disp('On the coldest day only consumption at night, so NO STORAGE SATURATION on this day')
    Selection.bandera=0;
    Selection.Temperature_Limit=valm;
    Selection.Position=posm;
else Selection.bandera=1;
end

end
```

4) Model of Daily Consumption

This program obtains the linear model approximation of each houses and performs an error distribution study. Combining both, each house is modelled.

Contents

Model of Daily Consumption

This program models the daily consumption by using metered data

```
start_time=22;
end_time=7;


long=length(Temp(:,1));
long2=length(Temp(:,1))-24;
Y=1:24:long+1;                  %divide the year in days
for i=1:length(Y)-2
     av_temp(i,1)=sum(Temp(Y(i)+start_time-1:Y(i+1)+start_time-1,2))/24;
    %required heating hours
end;
% This program computes heat_degree_hours strating from the '01-Jan-2012 22:00'.
```

Average House

```
%input('group of study:  ');
reg=input('group of study, depending on storage %: ');
num=input('number of house (0 if AVERAGE HOUSE): ');
fil=length(reg(:,1));
if num==0 col=length(reg(1,:));
    for i=1:col
        X=reg(:,i);
        Average_house(i)=mean(X);
    end
else Average_house(1,:)=reg(num,:);
end
%Consumption of this Average house in ToU(from 22 to 46 hours THE FIRST)
for i=1:length(Y)-2
    daily_storeh(i,1)=trapz(Average_house(Y(i)+start_time-1:Y(i+1)+end_time-1));
    %integrate of the whole day from 22:00 to 31:00 on the first round
end;
```

Regression to check Linnear Model.

```
longreg=length(daily_storeh);
regres=[daily_storeh(2:longreg),av_temp(1:longreg-1)];
XX=regres(:,2);
YY=regres(:,1);
```

Compute the error and see distribution hourly

```
long=length(YY);
Y1=daily_storeh(2:longreg);
hourly_storeh=daily_storeh/24;
YY=hourly_storeh(2:longreg);

mdl = LinearModel.fit(XX,YY);
A_regres=mdl.Coefficients.Estimate(1);
B_regres=mdl.Coefficients.Estimate(2);
Temperature=Temp(:,2);
A=A_regres*sum(Average_house)*0.84/sum(daily_storeh);  % Corrected for real cons level
B=B_regres*sum(Average_house)*0.84/sum(daily_storeh);
Estimation=B*Temperature+A;     % Linear estimation;
```

Error Distribution Study & Plotter

```
error=mdl.Residuals.Raw;   %Estimation-metered value

% Normality tests
flag1=lillietest(error);
flag2=jbtest(error);
flag3=kstest(error);
flag4=ttest(error);

% Creation of error bounds
err_lim=max(abs(error));
inf_limit=-err_lim;
sup_limit=err_lim;
% Generation of 7 levels of deviation
sector_length=sup_limit*2/6;
dev_lev=inf_limit:sector_length:sup_limit;
dev_points_prob=inf_limit-(sector_length/2):sector_length:sup_limit+(sector_length/2);

disp('-----------------------------------------------------')
X=[' Linear model: q(T)= ',num2str(B),'*T+',num2str(A)];
disp(X)
if (flag1&flag2&flag3)==0  %Normality of residuals

% Plot data on the left and display statistics and test results on right
fig(1) = figure('Name', 'Normal Distribution');
currData =error;   %where residuals are stored
currSort = sort(currData);
% Use histfit to plot a histogram of the data overlaid with a normal
% distribution based on its mean and standard deviation
histfit(currData,30);
title(sprintf('Normally distributed data, %d samples',length(currData)));
disp(' Residuals: Normality test passed');
phat = mle(error,'distribution','Normal');
cum_prob = cdf('Normal',dev_points_prob,phat(1,1),phat(1,2));
for i=2:length(cum_prob)
    probabilities(1,i-1)=cum_prob(1,i)-cum_prob(1,i-1);
end
Norm=1;

elseif flag4==0   %T-Student
fig(1) = figure('Name', 'T-Student Distribution');
currData =error;   %where residuals are stored
currSort = sort(currData);
% Use histfit to plot a histogram of the data overlaid with a normal
% distribution based on its mean and standard deviation
histfit(currData,30,'tlocationscale');
title(sprintf('T-Student distributed data, %d samples',length(currData)));
disp(' Residuals: T-Student distribution');
phat = mle(error,'distribution','tlocationscale');
cum_prob = cdf('tlocationscale',dev_points_prob,phat(1,1),phat(1,2),phat(1,3));
for i=2:length(cum_prob)
    probabilities(1,i-1)=cum_prob(1,i)-cum_prob(1,i-1);
end
Norm=0;
end

disp('-----------------------------------------------------')
```

5) Stochastic Genetic Algorithm

Contents

Stochastic Genetic Algorithm for storage level simulation

```
inf=day;
sup=inf+0;
lb=zeros(1,24);
ub=ones(1,24);
intCon=1:24;
Elspot=Elspot_Days(inf,:);
Y=1:24:17569;
if pred==1 figure('Name','Stochastic GA steps. FORECAST & SIMULATION');
else figure('Name','Stochastic GA steps. SIMULATION WITH REAL TEMPERATURE');
end

S_0=S_0_simulated;
```

Application of Model developed in Model_Daily_Cons

```
for j=1:7
    for i=1:24
        val=Estimation2(1,i)+dev_lev(1,j);    %forecast for the selected day
        if val<=0 q(i,j)=0;
        else q(i,j)=val;     % SGA uses q based on Estimation2
        end
    end
end
unos=floor(sum(Estimation2)*0.84/P)+1;     %minimum number of charging hours

for dev=1:7
```

First Generation Response Vectors

```
x1=lb;
cut_p=mean(Elspot);
for k=1:length(Elspot)
    price=Elspot(k);
    if price<cut_p x1(k)=1;
    end
end
% Storage Simulation:
    S_rem=zeros(1,24);
    S_rem(1)=S_0;
    for j=1:23
        S_rem(j+1)=S_rem(j)+P*x1(j)-q(j,dev);
    end
```

Mutation

```
    S_0=S_0_simulated;
    [x_mut S_rem S_0_next]=mutation(x1,S_0,C,P,q,dev);
    Storeh_evol_dev(dev,:)=S_rem;
    for k=1:24
        valor=x_mut(k)*Elspot(k);
        cost(k)=valor;
    end
    costs=sum(cost);
    earned_mut=sum(x_mut)*Price_tariff-costs;  %selling-incomes (tariff)
    x_7_mut(dev,:)=x_mut;

subplot(3,3,dev)
stairs(Elspot,'g')
hold on
stem(x_7_mut(dev,:)*50,'r')
plot(q(:,dev)*10,'m')
plot(med_temp(:),'c')
S_percentaje=S_rem*100/C;
plot(S_percentaje,'-+')   %only today's remaining level
tit=['SGRV control, Deviation:',num2str(dev_lev(dev))];
title(tit)
```

```
end
```

Genetic Algorithm's application in Natural Selection

First generation

```
num_pop=round(probabilities*1000);
pop=1;
k=1;
for j=1:7
    for i=1:num_pop(j)
        first_gen(k,:)=x_7_mut(j,:);
        k=k+1;
    end
end
dev=4;    % THE AVERAGE CONSUMPTION IS ON THE dev=0, NUMBER 4.
```

GA in Matlab

```
opts=gaoptimset('PopulationSize',k,'InitialPopulation',first_gen);
[xOpt,fval,exitflag,output,population]=ga(@(x)fun(x,Elspot),24,...
        [],[],[],[],lb,ub,@(x)const_7_step(x,S,q,S_0,P,unos,dev),intCon,opts);
% xOpt=round(xOpt);          %CORRECTION TO 1 AND 0  NO NEEDED
[xOpt_mut S_rem ok_lev]=mutation_7_step(xOpt,S_0,S,P,q,unos,dev);

second_gen=[xOpt;xOpt_mut];  %Second generation:

% Repetition until satisfying both
```

Storage Simulation

```
S_0=S_0_simulated;
for i=1:23
   sto=S_0+P*xOpt(i)-q(i,dev); %storeh level
   if sto<0 xOpt(i+1)=1;     %CHARGE IT!
   elseif sto>C xOpt(i+1)=0;  %too much, DON'T CHARGE IT!
   end
   S_rem_ga(i)=S_0+P*xOpt(i)-q(i,dev);     %fill S_rem vector
   S_0=S_rem_ga(i);
end
S_rem_ga(24)=S_0+P*xOpt(24)-q(24,dev);
S_0_next=S_rem_ga(24);

subplot(3,3,8)
% plot(Storeh_evol_dev(dev,:))
hold on
stairs(Elspot,'g')
stem(xOpt*50,'r')
plot(q(:,dev)*10,'m')
plot(med_temp,'c')
S_percentaje=S_rem_ga*100/C;
plot(S_percentaje,'-+')    %only today's remaining level
title('GENETIC ALGORITHM RESULT')
```

Mutation of GA's result

To keep all the solution within the bounds

```
S_0=S_0_simulated;
[xOpt_mut S_ga_mut S_0_next]=mutation(xOpt,S_0,S,P,q,dev);
for k=1:24
    valor=xOpt_mut(k)*Elspot(k);
    cost_SGA(k)=valor;
end
    cost_total_SGA=sum(cost_SGA);
```

Displays

```
disp('---------------------------------------------------------')
disp('           End of STOCHASTIC GENETIC ALGORITHM                 ')
disp('---------------------------------------------------------')
EARNED=(sum(xOpt_mut)*Price_tariff-costs)*P/1000;  %selling-incomes (tariff)
X=['Money Earned with this solution: ',num2str(EARNED),' €'];
disp(X)
X=['Minimum charging hours needed: ',num2str(unos)];
disp(X)
X=['Number of hours offered by this solution: ',num2str(sum(xOpt_mut))];
disp(X)
X=['Storage level for tomorrow: ',num2str(100*S_rem(24)/S),'%'];
disp(X)
X=['Heat Consumption plan: ',num2str(sum(xOpt_mut)*P),' kWh'];
disp(X)
disp('----------------------------------------------------------')
subplot(3,3,9)
hold on
plot(Elspot,'g')
stem(xOpt_mut*50,'r')
plot(q(:,dev)*10,'m')
plot(med_temp,'c')
S_percentaje=S_rem*100/C;
plot(S_percentaje,'-+')   %only today's remaining level
title('GA MUTATED')
hold off
tightfig;
```

6) Fitness function inside GA

Target: minimize costs by using daily cheapest hours.

```
function y=fun(x,Elspot)
%energy I buy from Elspot
y=x(1)*Elspot(1)+x(2)*Elspot(2)+x(3)*Elspot(3)+x(4)*Elspot(4)+x(5)*Elspot(5)+...
    x(6)*Elspot(6)+x(7)*Elspot(7)+x(8)*Elspot(8)+x(9)*Elspot(9)+x(10)*Elspot(10)+...
    x(11)*Elspot(11)+x(12)*Elspot(12)+x(13)*Elspot(13)+x(14)*Elspot(14)+...
    x(15)*Elspot(15)+x(16)*Elspot(16)+x(17)*Elspot(17)+x(18)*Elspot(18)+...
    x(19)*Elspot(19)+x(20)*Elspot(20)+x(21)*Elspot(21)+x(22)*Elspot(22)...
    +x(23)*Elspot(23)+x(24)*Elspot(24);

end
```

7) Internal constrains inside GA

Number of charging hours [c(1)], bounds for initial level of tomorrow [c(2)] & [c(3)], Power Balance equation for the selected day [c(4)]

```
function [c ceq] = const(x,S,q,S_0,P,unos)

S_rem=zeros(1,24);
    val=S_0+P*x(1)-q(1)-S;
    c(1)=val;
    val2=S_0+P*x(1)-q(1);
    S_rem(1)=val2;
    for j=2:24
        val3=S_rem(j-1)+P*x(j)-q(j)-S;
        c(j)=val3;
        val4=S_rem(j-1)+P*x(j)-q(j);
        S_rem(j)=val4;
    end

c(2)=(x(1)+x(2)+x(3)+x(4)+x(5)+x(6)+x(7)+x(8)+x(9)+x(10)+x(11)+x(12)+...
    x(13)+x(14)+x(15)+x(16)+x(17)+x(18)+x(19)+x(20)+x(21)+x(22)+x(23)+...
    x(24))-unos;    %charging hours around 8 a day

cons=sum(q);

c(3)=cons-S_0-(x(1)+x(2)+x(3)+x(4)+x(5)+x(6)+x(7)+x(8)+x(9)+x(10)+x(11)+x(12)+...
    x(13)+x(14)+x(15)+x(16)+x(17)+x(18)+x(19)+x(20)+x(21)+x(22)+x(23)+...
    x(24))*P;  %Daily Power Balance inequation

ceq=[];   %THIS ONE MUST BE EMPTY, MATLAB CONSTRAIN

end
```

8) Capacity Plotter

Contents

Capacity Plotter

```
max_cons=WinterTime.Selection.maximo;
mini=WinterTime.Selection.minimo*1.1;

for i=1:7
X_grid(i)=Y(i)-1;
X_med(i)=Y(i)+8;
end
figure('Name','Day Limit and Capacity of the House');
subplot(2,1,1);
hold on
title('Cluster Tool application');
% Select the days to plot
if Selection.Day_Limit>3
    lower=Selection.Day_Limit-3;
    upper=Selection.Day_Limit+3;
elseif Selection.Day_Limit>1
    lower=Selection.Day_Limit-1;
    upper=Selection.Day_Limit+5;
else lower=Selection.Day_Limit;
    upper=Selection.Day_Limit+6;
end
```

Day Selector and plotter

```
plot(Average_house(1,Y(lower):Y(upper)),'Linewidth',1);
plot(Temperature(Y(lower):Y(upper)),'m');
stairs(X_grid,Annual_Cons.Average_Temp(lower:upper),'r','Linewidth',2);    % Average Temp
text(X_med(1:6),Annual_Cons.Average_Temp(lower:upper-1)-2,num2str(Annual_Cons...
    .Average_Temp(Selection.Day_Limit-3:Selection.Day_Limit+2),4),'Color',[1 0 0]);
Xc=['Capacity: ',num2str(C),' kWh'];
text(X_med(3),max_cons+2,Xc,'Linewidth',2);

grid on
set(gca,'xtick',X_grid(1:7));
```

Set the shape of Cluster Tool

```
for i=1:6
    val_ini=Y(i+1)-3; % start time
    val_fin=Y(i)+6;   % end time
    line([val_ini val_ini],[max_cons mini],'Color',[0 0 0],'Linewidth',2)  % Vertical
    line([val_fin val_fin],[max_cons mini],'Color',[0 0 0],'Linewidth',2)  % Vertical
    line([val_ini val_fin],[mini mini],'Color',[0 0 0],'Linewidth',2)  % Horizontal
end
line([0 Y(1)+6],[max_cons max_cons],'Color',[0 0 0],'Linewidth',2)  % Horizontal
line([Y(i+1)-3 144],[max_cons max_cons],'Color',[0 0 0],'Linewidth',2)  % Horizontal
for i=2:6
    val_fin=Y(i)+6; % start time
    val_ini=Y(i)-3;   % end time
    line([val_ini val_fin],[max_cons max_cons],'Color',[0 0 0],'Linewidth',2)  % Horizontal
end
```

Annual Heat Percentajes

```
subplot(2,1,2);
hold on
title('Heat Storage Per100 along the year and Capacity Selector');
grid on
set(gca,'ytick',0:0.1:1);
plot(Annual_Cons.Heat_per100,'r')
stem(Selection.Day_Limit,Annual_Cons.Heat_per100(Selection.Day_Limit),'Linewidth',2);
Xc=['Capacity: ',num2str(C,4),' kWh'];
text(Selection.Day_Limit,Annual_Cons.Heat_per100(Selection.Day_Limit)...
    +0.05,Xc,'EdgeColor','b','LineWidth',2);
```

9) Control Application and Display

Contents

Application of the Final Response Vector to the household and final response.

in "Estimation2" there is registered the forecasted consumption from Estimation2; in "xOpt_mut" there is the response vector given by the SGA. in "q_run", the consumption based on Real Temperature. in "S_real" there is the evolution when applying the SGA suggestion. in "med_cons" there is the 24-hour temperature forecast. S_rem out of SGA is the latest one (GA solution)

```
x_SGA=xOpt_mut;
q_real;   % from ToU simulation
figure('Name','Application of the solution to Real Temperature');
subplot(1,2,1);
S_real=zeros(1,24);   %initial value of vector
x_final=xOpt_mut;     %real vector used by the customer

S_real(1)=S_0_REAL;
for i=1:23
    sto=S_real(i)+P*x_final(i)-q_real(i); %storeh level
    if sto<0 x_final(i)=1;    %CHARGE IT!
    elseif (sto>0)&(sto<0.25*S) x_final(i)=1; %below 25%, CHARGE IT!
    elseif sto>S x_final(i)=0;  %too much, DON'T CHARGE IT!
    end
    S_real(i+1)=S_real(i)+P*x_final(i)-q_real(i);    %fill S_rem vector
               %CORRECTED BY THE TANK
end

S_initial_tomorrow=S_real(24)+P*x_final(24)-q_real(24);
S_rem_100=S_rem*100/C;    %Percentaje instead of real level, more intuitive
S_real_100=S_real*100/C;
plot(S_rem_100,'r+-');    % given by SGA
hold on
plot(Elspot,'g')
stem(x_final*50)
stem(xOpt_mut*25,'r:')
plot(q(:,4)*10,'r:')
plot(q_real*10,'--')
plot(Temperature(Y(day):Y(day+1)-1),'c')    %real temperature
plot(S_real_100,'-+')    %only today's remaining level
title('SGA application and system´s answer')
% Level study
subplot(1,2,2);
hold on
ejes=[0 24 -10 60];
stem(x_final*50)
stem(xOpt_mut*25,'r:')
axis(ejes)
plot(Elspot,'g:');
hold off
title('Control given in RED; real used in BLUE')
```

Control of capacity

```
v_100=(x_final~=xOpt_mut);   % look for the difference between vectors
k=0;
pos_change=zeros(2,24);
for i=1:24
    if (v_100(i)==1)&[(xOpt_mut(i)==1)&(x_final(i)==0)]
        %this value has been changed when SGA says "charge"
        k=k+1;
        % 100% reached by real solution
        pos_change(1,i)=-1;     % -1 tank doesn't charge
        pos_change(2,i)=i;      % give the position
    elseif (v_100(i)==1)&[(xOpt_mut(i)==0)&(x_final(i)==1)]
        %this value has been changed when SGA says "Don't charge"
        k=k+1;
        pos_change(1,i)=1;      % 1 tank charges
        pos_change(2,i)=i;      % give the position
    end
end
% pos_change registeres all the differences bewteen vetors.
pos=max(pos_change(2,:));
if pos==0;
    start_control=0;
elseif pos~=0
    change=pos_change(1,pos);
    start_control=max(pos);     % hour of the change
end
% The amount of heat needed to reach 100% is less than P (power) one hour
% before, reason why the control doesn't use that hour. Our storage is
% between (1-P/C)% and 100% (C) of its capacity. To be conservative,
% the value selected is (1-P/C)% and info in uploaded
% S_rem_100 and S_real_100;

S_control=zeros(1,24);  % adjustment in 2 days
for l=1:24
    S_control(l)=S_rem(l);   % data from SGA
end
% TWO INDICATORS: FULL STORAGE OR EMPTY STORAGE
if (pos~=0)&(change==-1) S_control(start_control)=C-P; % Translate to "Storage full"
    % Conservative gap, "-P" that doesn't charge
elseif (pos~=0)&(change==1) S_control(start_control)=0.2*C; % Translate to "Storage Empty"
    % Conservative gap, "20% is worst situation
end
% Update storage evolution from last change
if start_control~=0
    for k=start_control:23
            S_control(k+1)=S_control(k)+P*x_final(k)-q_real(k);
    end
end

% Show me the change done:
if (init==1)&(pos~=0)
    control_tool(pos)=S_control(start_control);
elseif (init==0)&(pos~=0)
    control_tool(pos+24)=S_control(start_control);
end
S_0_control_tomorrow=S_control(24)+P*xOpt_mut(24)-q_real(24);
% Level estimation for tomorrow based on the control performed
```

Plotter

```
if init==1
    S_control_2days=zeros(1,48);
    S_real_2days=zeros(1,48);
    x_final_2days=zeros(1,48);
    xopt_mut_2days=zeros(1,48);
    for k=1:24
    S_control_2days(k)=S_control(k);
    S_real_2days(k)=S_real(k);
    x_final_2days(k)=x_final(k);
    x_SGA_2days(k)=x_SGA(k);
    end
    % Plotter of control:
    figure('Name','Storage Level Control');
    X=['Day of Control: ',num2str(day)];
    title(X);
    hold on
    plot(S_control*100/C,'r+-')
    plot(Elspot,'g');
    plot(Temperature(Y(day):Y(day+1)-1),'c');
    plot(S_real*100/C,'+-')
    if pos~=0 stem(control_tool*100/C,'m:')    % Control starts
    end
    hold off
elseif init~=1
    for k=25:48
    S_control_2days(k)=S_control(k-24);
    S_real_2days(k)=S_real(k-24);
    x_final_2days(k)=x_final(k-24);
    x_SGA_2days(k)=x_SGA(k-24);
    end
    % Plotter of control:
    figure('Name','Storage Level Control');
    X=['Day of Control: ',num2str(day-1),' & ',num2str(day)];
    title(X);
    hold on
    plot(S_control_2days*100/C,'r+-')
    precio=[Elspot_Days(day-1,:) Elspot_Days(day,:)];
    plot(precio,'g');
    line([24 24],[-30 100],'Color',[0 0 0]);
    plot(Temperature(Y(day-1):Y(day+1)-1),'c');
    plot(S_real_2days*100/C,'+-')
    stem(x_final_2days*25);      %Used
    stem(x_SGA_2days*50,'r:');    %Given
    if pos~=0 stem(control_tool*100/C,'m:')    % Control starts
    end
    set(gca,'xtick',[1:1:48]);
    hold off
end
```

10) ARMA model

Forecast function

```
function [Yf,upper, lower] = ambforecast(Y,fit)

[Yf,YMSE] = forecast(fit,24,'Y0',Y);
upper = Yf + 1.96*sqrt(YMSE);
lower = Yf - 1.96*sqrt(YMSE);
end
```

```
i = day1;
[Tempridict(1,((i-1)*24+1):(i*24)),TempridictUP(1,((i-1)*24+1):(i*24)),...
    TempridictLOW(1,((i-1)*24+1):(i*24))] = ambforecast(Temper(1:(i-1)*24),fit);
```

Forecast Simulation

Contents

Temperature Forecast with ARMAX model and Consumption

```
Temper=Temp(:,2);

inf=inf-1;
num=inf*24;
day1=inf;
day2=inf+1;    %selection of past days for ARMAX

ModelData
PredictData
```

Plot of Real & Forecasted temperatures profile

```
plot(Tempridict,'DisplayName','Tempridict');hold all;plot(TempridictLOW,...
    'DisplayName','TempridictLOW'); plot(TempridictUP,'DisplayName','TempridictUP');hold off;
hold on
tttem=Temp(1:inf*24,2);
plot(tttem,'c:')

med_temp=Tempridict(1,((i-1)*24+1):(i*24));
Te=[Temper(1:num);med_temp'];
```

Back to the day

```
inf=day2;
```

11) Total Compilation of all programs & Simulation

Contents

Household's Total Simulation

Load all the customers given by the Company

```
init=input('Initiate Control? "1" yes": ');
if init==1
```

```
    Area=input('Area of study? JYVÄSKYLÄ=1; HELSINKI=2; TAMPERE=3; TAVASTIA=4: ');
    if Area==1
    load('CentralFinland_JYVASKYLA.mat');
    Temp=Temp_Jyvaskyla;
    load('Region_4')
    name=Region_4;
    elseif Area==2
    load('Uusimaa_HELSINKI.mat'); %input('Temperature´s profile: ');
    Temp=Temp_Helsinki;
    load('Region_1')
    name=Region_1;
    elseif Area==3
    load('Pirkanmaa_TAMPERE.mat');
    Temp=Temp_Tampere;
    load('Region_3')
    name=Region_3;
    elseif Area==4
    load('TavastiaProper_HAMEENLINNA.mat');
    Temp=Temp_Tavastia;
    load('Region_2')
    name=Region_2;
    end
```

Cluster Tool applied to a Region

```
[storeh_45_60 storeh_60_75 storeh_75_100]=Cluster_Tool(name);
```

Model of Daily Consumption

```
Model_Daily_Cons
```

LEGO Simulation

```
[C Temp_Limit Annual_Cons WinterTime Selection]=LEGO_Simulation(Average_house,num,Temp,1);
```

Capacity Plotter app

```
Capacity_Plotter;
```

Regression Model's application

```
% this part creates all the deviations based on the regression model
% from HISTORICAL DATA
% Average House Jyvaskyla simulation
    Temperature=Temp(:,2);
    load('Elspot_Days (732x24).mat')
    load('Price_tariff.mat')
```

Plot deviation levels:

```
for j=1:7
    for i=1:17568
        val=Estimation(i,1)+dev_lev(1,j);
        if val<=0 Q_dev(i,j)=0;
        else Q_dev(i,j)=val;
        end
    end
end

figure('Name','Deviated levels of consumption');
subplot(2,3,[1,2,4,5]);
plot(Q_dev,'DisplayName','Q_dev');
title(sprintf('Consumption Levels based on deviations'));
subplot(2,3,3);
set(gca, 'visible', 'off');
text(0,1,'Probability of each deviation');
rows=[0.2 0.3 0.4 0.5 0.6 0.7 0.8];
format short
for i=1:length(rows)
    X=['Deviation: ',num2str(dev_lev(i),3)];
    text(0,rows(i),sprintf(X));
    X=[' Probability: ',num2str(probabilities(i),3)];
    text(0.5,rows(i),sprintf(X));
end
```

Generation of a random storage level

```
a=0;
S_ToU_random=a + (C-a).*rand(1,1);
mu=phat(1,1);
sigma=phat(1,2);
```

Customer's behaviour Simulation

```
subplot(2,3,6)
if Norm==1
    for i=1:length(Estimation)
        R(i) = normrnd(mu,sigma);
        simulation(i)=Estimation(i)+R(i);
    end
    histfit(R,40,'Normal')
    title('Errors Simulated: 7-step Normal');

elseif Norm==0
    pd=makedist('tlocationscale','mu',phat(1,1),'sigma',phat(1,2),'nu',phat(1,3));
    for i=1:length(Estimation)
        T(i) = random(pd);
        simulation(i)=Estimation(i)+T(i);
        if simulation(i)<=0 simulation(i)=0;
        end
    end
    histfit(T,40,'tlocationscale')
    title('Errors Simulated: 7-step T-Student');
end
```

ToU Performance for Simulation

in this part, ToU is simulated to have a real storage behaviour. Later one of this days will be selected to apply SGA.

```
x1=[1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1];   % vector of Night Use
S=C;    % Capacity
P=S/9;      % Bolier Power
simulation=simulation';
long=length(Estimation);
Y=1:24:long+1;
unos=9;
S_0_ToU_next=S_ToU_random;

for i=1:length(Y)-1
 q_sim=simulation(Y(i):Y(i+1)-1);
 [x1_mut S_ToU_year ok_lev]=mutation_simul(x1,S_0_ToU_next,S,P,q_sim,1);

 S_0_ToU_next=S_ToU_year(24)+P*x1_mut(24)-q_sim(24,1);
% Update S_0 for the next day:
    Storeh_evol(Y(i):Y(i+1)-1)=S_ToU_year;
    Elspot_curve(Y(i):Y(i+1)-1)=Elspot_Days(i,:);
    x_year(Y(i):Y(i+1)-1)=x1_mut(:);
    x_day_by_day(:,i)=x1_mut;
    storeh_evol_year(:,i)=S_ToU_year;
end
```

Display

```
load('Elspot_Days (732x24).mat')
repres=input('Show ANNUAL Simulation based on REAL DATA? "1" YES: ');
if repres==1
all=input('the whole year? "1" yes, "0" no: ');
if all==1
    figure('Name','ToU Simulation along the whole time domain');
    plot(Storeh_evol*100/C,'-+')
    hold on
    plot(Elspot_curve,'g:')
    stem(x_year*50,'r:')
    plot(simulation*10,'m:')
    plot(Temp(:,2),'c:')
    title('SIMULATION WITH ToU')
    hold off
    day=input('select day to study: ');
else
    figure('Name','ToU behaviour');
    day=input('select day to show: ');
    plot(Storeh_evol(Y(day):Y(day+1)-1)*100/C,'-+')
    hold on
    Elspot=Elspot_Days(day,:);
    plot(Elspot,'g:')
    stem(x_day_by_day(:,day)*50,'r:')
    plot(q_sim*10,'m:')
    plot(Temp(Y(day):Y(day+1)-1,2),'c:')   %real temperature
    titul=['Real historical behaviour, Day ',num2str(day)];
    title(titul);
    hold off
end
else day=input('select day to study: ');
end
```

Daily analysis to guess starting level

```
if init==1
    S_0_REAL=Storeh_evol(Y(day));
    levi=input('Level chosen to start iterations (in %): ');
    S_0_simulated=S*levi/100;

else day=day+1;
    S_0_REAL=S_initial_tomorrow;
    S_0_simulated=S_0_control_tomorrow;
    disp('------- Day after Control -------')
    X=['Level guessed for tomorrow with REAL TEMPERATURE: ',num2str(S_0_REAL*100/S)];
    disp(X);
end

% Real consumption level stored on the simulation of real behaviour

q_real=simulation(Y(day):Y(day+1)-1);
```

Stochastic GA for Simulation

```
pred=input('Temperature Forecast? "1" yes, "0" no: ');
if pred==1
    inf=day;
    Forecast_simulation                 %Forecasting the temperature of tomorrow
else
    med_temp=Temp(Y(day):Y(day+1)-1,2)';
end

Estimation2=B*med_temp+A;       % Estimation out of linear model
clearvars S_rem x1 x
SGA_simulation
```

Results Validation app of Simulation

```
Results_Validation
```

12) Results Validation

Contents

Application of the Final Response Vector to the household and final response.

in "Estimation2" there is registered the forecasted consumption from Estimation2; in "xOpt_mut" there is the response vector given by the SGA. in "q_run", the consumption based on Real Temperature. in "S_real" there is the evolution when applying the SGA suggestion. in "med_cons" there is the 24-hour temperature forecast. S_rem out of SGA is the latest one (GA solution)

```
x_SGA=xOpt_mut;
q_real;   % from ToU simulation
figure('Name','Application of the solution to Real Temperature');
subplot(1,2,1);
S_real=zeros(1,24);   %initial value of vector
x_final=xOpt_mut;     %real vector used by the customer

S_real(1)=S_0_REAL;
for i=1:23
    sto=S_real(i)+P*x_final(i)-q_real(i); %storeh level
    if sto<0 x_final(i)=1;    %CHARGE IT!
    elseif (sto>0)&(sto<0.25*S) x_final(i)=1; %below 25%, CHARGE IT!
    elseif sto>S x_final(i)=0;  %too much, DON'T CHARGE IT!
    end
    S_real(i+1)=S_real(i)+P*x_final(i)-q_real(i);    %fill S_rem vector
               %CORRECTED BY THE TANK
end

S_initial_tomorrow=S_real(24)+P*x_final(24)-q_real(24);
S_rem_100=S_rem*100/C;    %Percentaje instead of real level, more intuitive
S_real_100=S_real*100/C;
plot(S_rem_100,'r+-');   % given by SGA
hold on
plot(Elspot,'g')
stem(x_final*50)
stem(xOpt_mut*25,'r:')
plot(q(:,4)*10,'r:')
plot(q_real*10,'--')
plot(Temperature(Y(day):Y(day+1)-1),'c')   %real temperature
plot(S_real_100,'-+')   %only today's remaining level
title('SGA application and system´s answer')
```

Level study

```
subplot(1,2,2);
hold on
ejes=[0 24 -10 60];
stem(x_final*50)
stem(xOpt_mut*25,'r:')
axis(ejes)
plot(Elspot,'g:');
hold off
title('Control given in RED; real used in BLUE')
```

Control of capacity

```
v_100=(x_final~=xOpt_mut);    % look for the difference between vectors
k=0;
pos_change=zeros(2,24);
for i=1:24
    if (v_100(i)==1)&[(xOpt_mut(i)==1)&(x_final(i)==0)]
        %this value has been changed when SGA says "charge"
        k=k+1;
        % 100% reached by real solution
        pos_change(1,i)=-1;      % -1 tank doesn't charge
        pos_change(2,i)=i;       % give the position
    elseif (v_100(i)==1)&[(xOpt_mut(i)==0)&(x_final(i)==1)]
        %this value has been changed when SGA says "Don't charge"
        k=k+1;
        pos_change(1,i)=1;       % 1 tank charges
        pos_change(2,i)=i;       % give the position
    end
end
% pos_change registeres all the differences bewteen vetors.
pos=max(pos_change(2,:));
if pos==0;
    start_control=0;
elseif pos~=0
    change=pos_change(1,pos);
    start_control=max(pos);      % hour of the change
end
% The amount of heat needed to reach 100% is less than P (power) one hour
% before, reason why the control doesn't use that hour. Our storage is
% between (1-P/C)% and 100% (C) of its capacity. To be conservative,
% the value selected is (1-P/C)% and info in uploaded
% S_rem_100 and S_real_100;

S_control=zeros(1,24);    % adjustment in 2 days
for l=1:24
    S_control(l)=S_rem(l);    % data from SGA
end
% TWO INDICATORS: FULL STORAGE OR EMPTY STORAGE
if (pos~=0)&(change==-1) S_control(start_control)=C-P; % Translate to "Storage full"
    % Conservative gap, "-P" that doesn't charge
elseif (pos~=0)&(change==1) S_control(start_control)=0.2*C; % Translate to "Storage Empty"
    % Conservative gap, "20% is worst situation
end
% Update storage evolution from last change
if start_control~=0
    for k=start_control:23
            S_control(k+1)=S_control(k)+P*x_final(k)-q_real(k);
    end
end
% Show me the change done:
if (init==1)&(pos~=0)
    control_tool(pos)=S_control(start_control);
elseif (init==0)&(pos~=0)
    control_tool(pos+24)=S_control(start_control);
end
S_0_control_tomorrow=S_control(24)+P*xOpt_mut(24)-q_real(24);
% Level estimation for tomorrow based on the control performed
```

Plotter

```
if init==1
    S_control_2days=zeros(1,48);
    S_real_2days=zeros(1,48);
    x_final_2days=zeros(1,48);
    xopt_mut_2days=zeros(1,48);
    for k=1:24
    S_control_2days(k)=S_control(k);
    S_real_2days(k)=S_real(k);
    x_final_2days(k)=x_final(k);
    x_SGA_2days(k)=x_SGA(k);
    end
    % Plotter of control:
    figure('Name','Storage Level Control');
    X=['Day of Control: ',num2str(day)];
    title(X);
    hold on
    plot(S_control*100/C,'r+-')
    plot(Elspot,'g');
    plot(Temperature(Y(day):Y(day+1)-1),'c');
    plot(S_real*100/C,'+-')
    if pos~=0 stem(control_tool*100/C,'m:')    % Control starts
    end
    hold off
elseif init~=1
    for k=25:48
    S_control_2days(k)=S_control(k-24);
    S_real_2days(k)=S_real(k-24);
    x_final_2days(k)=x_final(k-24);
    x_SGA_2days(k)=x_SGA(k-24);
    end
    % Plotter of control:
    figure('Name','Storage Level Control');
    X=['Day of Control: ',num2str(day-1),' & ',num2str(day)];
    title(X);
    hold on
    plot(S_control_2days*100/C,'r+-')
    precio=[Elspot_Days(day-1,:) Elspot_Days(day,:)];
    plot(precio,'g');
    line([24 24],[-30 100],'Color',[0 0 0]);
    plot(Temperature(Y(day-1):Y(day+1)-1),'c');
    plot(S_real_2days*100/C,'+-')
    stem(x_final_2days*25);     %Used
    stem(x_SGA_2days*50,'r:');   %Given
    if pos~=0 stem(control_tool*100/C,'m:')    % Control starts
    end
    set(gca,'xtick',[1:1:48]);
    hold off
end
```